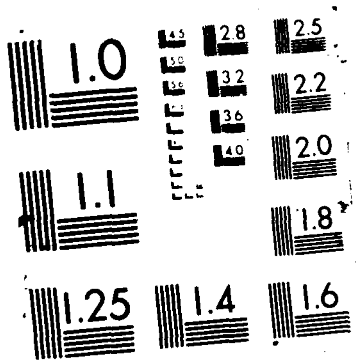
AD-A185 928 COURSE SCHEDULING AND OFFICER ASSIGNMENTS IN THE UNITED STATES MARINE CORPS: A HEURISTIC MODEL(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA K CHENG SEP 87 1/1

UNCLASSIFIED F/G 12/4 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COURSE SCHEDULING AND
OFFICER ASSIGNMENTS IN THE
UNITED STATES MARINE CORPS:
A HEURISTIC MODEL

by

Keng-Seng Chng

September 1987

Co-Advisor Richard E. Rosenthal
Co-Advisor Paul R. Milch

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; Distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable): 55	7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
COURSE SCHEDULING AND OFFICER ASSIGNMENTS IN THE UNITED STATES MARINE CORPS: A HEURISTIC MODEL

12 PERSONAL AUTHOR(S) CHNG, Keng Seng

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year Month Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1987 September	84

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Heuristic Model, course scheduling, officer assignment, GAMS Modeling System, Linear Programming

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A heuristic model is proposed to solve the officer course scheduling and assignment problem in the United States Marine Corps. This model divides the problem into two sub-problems, namely course scheduling and officer assignments. Each sub-problem is solved through a separate model formulation.

The course scheduling model uses a FORTRAN 77 implementation of a new heuristic. The officer assignment model is a linear program that is formulated and solved using the GAMS Modeling system. Both models run on an IBM 3033AP mainframe and on personal computers using the DOS operating system.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL: Prof. R. E. Rosenthal & Prof. P. R. Milch	22b TELEPHONE (Include Area Code): 408-646-2795; 22c OFFICE SYMBOL: 55R1 & 55Mh

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

Block 19. Abstract (cont)

The models were tested using FY 88 planning data supplied by Headquarters Marine Corps (HQMC). Results from test runs, each carrying a different assumption about HQMC's policy on officer assignments, indicate a clear improvement in course waiting time over past years. Using the model, the average waiting time for an officer ranges from 1.1 to 2.3 weeks, depending on the assumptions made. In the past, average waiting time has been greater than five weeks.

Course Scheduling and Officer Assignments in the United States Marine Corps: A Heuristic Model

by

Keng-Seng Chng
Major, Republic of Singapore Air Force
B.A.(Hons), University of Leeds, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

Author: Keng-Seng Chng

Approved by: Richard E. Rosenthal, Co-Advisor

Paul R. Milch, Co-Advisor

Donald R Barr, for
P. Purdue, Chairman,
Department of Operations Research

James M. Fremgen,
Acting Dean of Information and Policy Sciences

ABSTRACT

A heuristic model is proposed to solve the officer course scheduling and assignment problem in the United States Marine Corps. This model divides the problem into two sub-problems, namely course scheduling and officer assignments. Each sub-problem is solved through a separate model formulation.

The course scheduling model uses a FORTRAN 77 implementation of a new heuristic. The officer assignment model is a linear program that is formulated and solved using the GAMS Modeling system. Both models run on an IBM 3033AP mainframe and on personal computers using the DOS operating system.

The models were tested using FY 88 planning data supplied by Headquarters Marine Corps (HQMC). Results from test runs, each carrying a different assumption about HQMC's policy on officer assignments, indicate a clear improvement in course waiting time over past years. Using the model, the average waiting time for an officer ranges from 1.1 to 2.3 weeks, depending on the assumptions made. In the past, average waiting time has been greater than five weeks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

For their invaluable contribution towards this thesis, I would like to thank:

- the officer staff of the Manpower Department, Headquarters Marine Corps;
- Dr Alexander Meeraus and his associates at The World Bank for providing the use of a courtesy copy of the GAMS Modeling System;
- my advisors, Professors Richard E. Rosenthal and Paul R. Milch for their guidance;
- my wife, Seok Leng, for providing the morale sustenance.

I. INTRODUCTION

The training of a United States Marine Corps (USMC) officer has many phases. During the initial skill training phase, an officer attends a series of courses leading to his primary Military Occupational Speciality (MOS). The purpose of this thesis is to develop a suitable model for planning course schedules and officer assignments in the initial skill training phase. The opening chapter defines the problem and lays the outline for the remaining chapters of the thesis.

A. PROBLEM STATEMENT

The following features will be described to explain the problem:

(1) Officer groupings;

(2) Officer training tracks;

(3) Training schools;

(4) On-the-job training and leave requirements;

(5) Planning objective.

1. Officer Groupings

There are three ways to group USMC officers undergoing initial skill training:

(a) *Newly recruited rechanneled officers.* The officers sent for training into a primary MOS can be divided into newly recruited officers and officers rechanneled from other USMC MOS's. Rechannelling occurs for a number of reasons. First, an officer can volunteer for a lateral movement to another MOS. Second, the USMC can direct certain groups of officers to move laterally into other MOS's when there are critical shortages. A third reason is that some Air officers are grounded and reclassified into ground jobs. Finally, officers recruited under the "Intended MOS" scheme are contractually bound to rechannel into a new MOS after spending a fixed period in their first MOS.

(b) *Restricted Unrestricted Officers.* In general, restricted officers have a more limited scope of duties compared to their unrestricted counterparts. The term restricted officer will be treated in this thesis as being synonymous to the term warrant officer(WO).

(c) *Air Ground officers.* Of the 23 primary MOS's in the USMC, two are classified as Air and the remaining 21 as Ground MOS's.

2. Officer Training Tracks

The list of courses attended by an officer during the initial skill training phase is called the officer's *training track* for that phase. The courses forming a given

training track depend on four factors--first, the officer's primary MOS; second, whether he is a rechanneled or newly recruited officer; third, his officer entry program and finally, whether he is an unrestricted or warrant officer. The first factor specifies the primary MOS course to which the officer is to be sent. The second factor determines whether he has to undergo The Basic School (TBS) course. Newly recruited officers must attend the TBS course before they can start their primary MOS course. Rechanneled officers proceed directly to the latter course. Third, a newly recruited officer's entry program determines whether he has to attend the Officer Candidate School (OCS) course before starting at TBS. Officers under the Officer Candidate Course (OCC) entry program must attend the OCS course. Other officers are not required to do so. Fourth, a warrant officer will have a different training track from an unrestricted officer with the same MOS. The warrant officer will first of all attend a separate TBS class from the unrestricted officer. After TBS graduation, depending on the MOS, the warrant officer may either attend a separate MOS class that is reserved for officers from his group or he could attend a class that is mixed with unrestricted officers.

3. Training Schools

The USMC controls the Officer Candidate School, The Basic School and some MOS schools. Other MOS schools are controlled by the three other military services. The flexibility with which the USMC can schedule classes and assign officers to them vary, depending on whether the school is USMC-controlled, and on the school itself. For non-USMC controlled schools, the USMC receives an annual allocation of course seats from the conducting agency. The allocation shows course timings and places available for USMC candidates. It is difficult to change the allocations after they are made.

Among USMC-controlled schools, there are differences in the constraints affecting class size, class composition (between unrestricted and warrant officers) and the method of scheduling classes. OCS must have three classes scheduled per year. Each class has a duration of 10 weeks with no overlapping classes allowed. A class must have between 100 to 150 students. The passing rate for an OCS class is 55%. Within these constraints, the USMC has the freedom to select start dates for OCS classes.

Every year, there must be nine TBS classes, eight for unrestricted officers and one for WO's. Each unrestricted officer TBS class must have between 150 to 250

officers. There are no size limits for the one restricted officer TBS class. Because the USMC wants to synchronize TBS class start dates with the expected trainee availability pattern, it does not have much leeway in deciding when TBS classes are to be held. In modeling the problem, the TBS course schedule is assumed to be fixed. The USMC also has a policy for each TBS class to assign a minimum number of officers to designated MOS's. The number varies between MOS's, but is the same for all TBS classes.

At USMC-controlled MOS schools, the number of classes held per year depends on the school's output requirements. It will change from year to year since the annual output requirements are not fixed. Classes conducted by the schools must observe prescribed lower and upper class size limits. Also, apart from the Infantry school, all USMC controlled MOS schools do not have the capacity to conduct overlapping classes.

4. **On the Job Training and Leave Requirements**

For some MOS's, officers must be sent to a period of on-the-job training (OJT) before they can commence the MOS course proper. The training lasts for 12 weeks. Except for Infantry officers, all officers from MOS's without OJT are given two weeks leave after TBS. Infantry officers proceed to the earliest available MOS course after TBS since the Infantry training school is on the same base as TBS.

5. **Planning Objective**

Every year, Headquarters (HQMC) specifies a quota for assignments into each MOS. The planning objective is to schedule USMC courses in a manner such that these quotas and all constraints are met, and to assign officers to classes so as to minimize total course waiting time. For this problem, total course waiting time is measured by summing the unoccupied period between courses for every officer under training. It does not include the time spent on OJT or enforced leave after TBS graduation.

At present, the tasks of course scheduling and officer assignment are handled by different organizations at HQMC. The former is the responsibility of the Training Department and the latter comes under the Manpower Department. Both agencies employ manual procedures. There is also no global model available for overall coordination between the two departments. The job is extremely laborious and involves drafting of initial plans, followed by adjustments for last-minute changes that inevitably occur. More important, because of the many complicating factors that must

be taken into account, it is unlikely these procedures could solve the problem to yield an optimal result.

B. PROBLEM SCOPE

The goal of this thesis is to develop a computer software package that HQMC could use to routinely solve the course scheduling and assignment problem. The model is based on a planning horizon of one year.

To limit its size, only courses and assignments involving unrestricted Ground officers are determined by the model. The variables for other officers will be fixed exogeneously by HQMC.

The model addresses a dynamic situation by taking into consideration events from the previous year. For USMC controlled MOS schools with no class overlaps, currently scheduled classes will not overlap with classes scheduled during the previous year. Also, there may be TBS graduates from the previous year who have not attended MOS schools. Provision have to be made to assign them to MOS classes held in the current FY.

C. MODEL SIMPLIFICATIONS

The model can be simplified by recognizing that it is possible to parametrize certain decision variables (in the original problem) without affecting optimality. The variables concerned are the OCS class start dates and number of Ground OCC officers assigned to an OCS class. First, OCS class dates can be selected so that each finishes at the time when an unrestricted officer TBS class is scheduled to begin. The dates selected must be such that no two classes overlap. There are usually several ways to construct such a course schedule. The next step is to calculate the number of Ground OCC officers to enter each OCS class. OCS classes consist solely of Ground and Air OCC officers and the number of Air OCC officers per class is fixed. Since the size for an individual OCS class is allowed to vary between 100 and 150, there is a range of 50 possible numbers from which the number of Ground OCC officers for each class can be selected. Choosing OCS class dates and Ground OCC officers per class in this manner will not affect optimality of the solution since the result obtained will show zero waiting time for all officers assigned from OCS to TBS classes. There is no better way to select these decision variables.

D. THESIS OUTLINE

There are three more chapters in this thesis. Chapter Two discusses different solution approaches that have been considered and explains details of the heuristic model proposed to solve the problem. In Chapter Three, the results from experimental runs are discussed. The test data were provided by HQMC and resemble closely the actual inputs used for FY 88 planning. Finally, the conclusion is covered in Chapter Four.

II. SOLUTION METHODOLOGY

The first section of this chapter examines the approaches which could be used to solve the problem. Sections two and three describe the proposed heuristic model.

A. SOLUTION APPROACHES

Three different approaches can be employed to solve the problem--*optimization, heuristics* and *simulation*. The last approach was not examined in depth by this study and interested readers are directed to an article by Plotnicki and Garfinkel [Ref. 1] describing how the course scheduling problem for an academic institution was solved using simulation.

Optimization covers those algorithms which provide solutions that are either guaranteed to be optimal or are within some certifiable, acceptable bound of optimality. Solution techniques not belonging to it are grouped loosely under heuristics. In real world problems, the discovery and implementation of computationally effective optimization algorithms are often difficult and sometimes impossible. Successful algorithms, if found, usually must be customized for specific problems.

Three optimization techniques were considered--*mixed integer programming* (*MIP*), *Lagrangean relaxation* and *Benders decomposition*. Details of how they could be applied to the Marine Corps problem are covered in Appendix A. An unsuccessful attempt was made to solve a simplified formulation of the problem using MIP on an IBM 3090 mainframe computer at The World Bank. No attempt was made to implement either of the two remaining optimization techniques on a computer because the structure and scale of the problem made it extremely difficult to develop a practical implementation. Instead, a heuristic model which made the problem computationally feasible was adopted for implementation.

The proposed heuristic model divides the problem into two parts. First, courses with discretionary course dates are scheduled using a heuristic algorithm. With course dates fixed, the second part solves the officer assignment problem through optimization. The break up of the problem into two smaller problems is aimed at developing solutions to the two smaller problems which can be merged to provide a good feasible solution for the overall problem. The first problem will be more difficult

to solve. Without considering the second problem, it must develop a set of course schedules which enables the officer assignments required for total waiting time minimization to be made. The model used to solve the first problem will be referred to as *the course scheduling model* and that for the second problem will be called *the officer assignment model.* The generic term used to describe the two models combined is *the heuristic model.*

B. COURSE SCHEDULING MODEL

This section explains the development of the course scheduling model. The same index and variable notation will be used throughout the section, that is:

Indices:

i denotes TBS class number

j denotes MOS class number

k denotes type of MOS course

Variables:

TBSEND(i)	is the end date of TBS class i
For a given MOS course,	
Class(j)	is the start date[1] of class j
n	is the number of classes scheduled per year
OJT	is the duration of on-the-job training before the MOS course
LV	is the duration of enforced leave before the MOS course
c	is the minimum number of officers that must be assigned to the MOS from each TBS class
ml	is the lower class size limit for every MOS class
mu	is the upper class size limit for every MOS class
q	is the annual MOS output quota

1. Constraints

There are five problem constraints which affect the course scheduling model:

(a) *Class capacity limits must be strictly enforced.* Every TBS class for unrestricted officers must have between 150 to 250 officers. No limits are imposed on the TBS class for warrant officers. Classes for USMC controlled MOS courses also have specified class size limits.

(b) *All USMC MOS schools except the Infantry school cannot conduct concurrent classes.*

[1] Time for this model will be measured in weeks.

(c) *Planning is based on a yearly cycle.* Therefore, the schedule for each MOS course must complete all classes within a fifty two week period.

(d) *Warrant officers must be sent to a TBS class separate from unrestricted officers.* In addition, only certain MOS courses will allow unrestricted and warrant officers to be assigned into the same MOS class.

(e) *Finally, unrestricted officer TBS classes must assign a specified minimum number of officers into certain MOS courses.*

2. Fundamental Concepts

Three concepts are introduced to present the proposed course scheduling model. The first concept is that of a "follow-on" MOS class to a given TBS class. For a given MOS course, MOS class j is termed a follow-on MOS class to TBS class i if class j is the first class of the MOS course to begin after TBS class i ends. An example will help to clarify this concept. Assume the MOS course has two classes taught per year. The scheduled dates for these two classes (i.e. C1 and C2) and the eight TBS classes (i.e. T1 to T8) are shown in the Figure 2.1 :

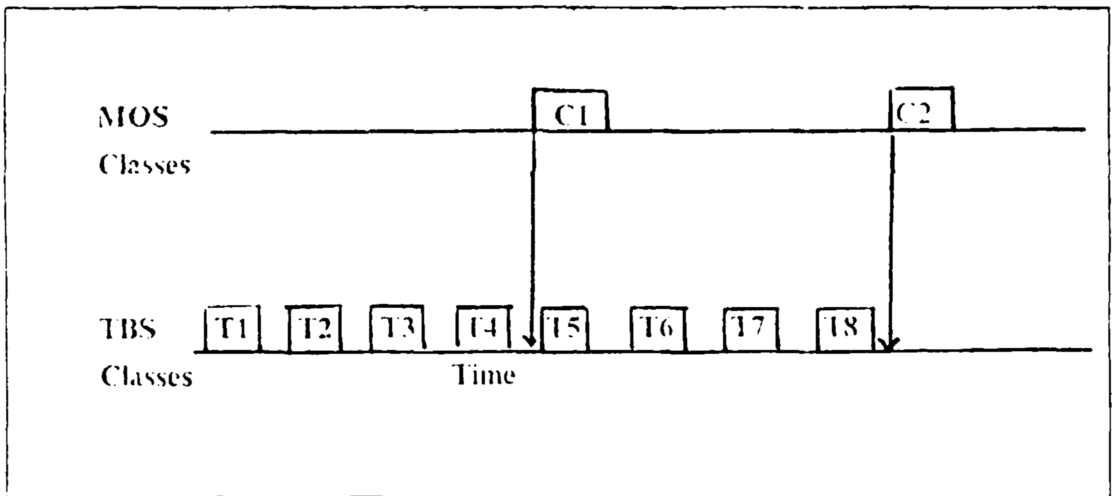


Figure 2.1 Follow-on and Back-to-Back classes.

In this example, MOS class one is the follow-on MOS class for TBS classes one, two, three and four. The second MOS class is the follow-on MOS class for the four remaining TBS classes. By definition, every TBS class will have a follow-on MOS class.

The second concept is related to the first. For a given MOS course, MOS class j is termed as a back-to-back class with TBS class i if the following conditions hold :

(a) MOS class j is a follow-on class for TBS class i

(b) TBS class i is the last TBS class to end before MOS class j begins.

Using the previous example, the two pairs of back-to-back classes are TBS class four with MOS class one and TBS class eight with MOS class two.

The final concept is an accounting variable called *TBSSIZE*(i). This variable measures the number of officers in TBS class i who are assigned to the MOS courses that have already scheduled as the algorithm progressively considers each MOS course. There are many ways to compute the *TBSSIZE*(i) values, each giving rise to a different answer. It will be shown in the next section how the proposed course scheduling model uses a heuristic method to decide which of these ways is to be chosen.

3. **Heuristic Principles**

The proposed model develops the schedules for USMC controlled MOS courses one at a time. Five heuristic principles are combined to develop each individual course schedule.

The first heuristic principle is used to compute the value n, the number of times the MOS course is taught in the year. Because of the restriction on a MOS class size, n must lie between the values q/mu and q/ml. The heuristic principle sets n to be equal to *Floor* (q/n) where *Floor* (x) represents the largest integer less than or equal to x.

The second principle says to spread the MOS classes so that there is an even distribution (over time) of back-to-back MOS classes with TBS classes[2]. This principle is motivated by the requirement for each TBS class to assign at least some officers to an MOS. In general, each MOS course has fewer classes than there are TBS classes. This makes it impossible to assign all officers entering the MOS into back-to-back MOS classes. By spreading the MOS classes "evenly", the aim is to minimize the maximum delay encountered by officers assigned to non back-to-back MOS classes.

The third heuristic principle states that whenever possible, back-to-back MOS classes should be scheduled to begin as early as possible, i.e., when their corresponding back-to-back TBS classes end[3].

---

[2] Usually, the number of warrant officers for any of the MOS's is small relative to unrestricted officers. Because of this, the second heuristic principle will be extended to exclude the scheduling of a back-to-back MOS class for the warrant officer TBS class.

[3] If officers have to be sent for on-the-job training or enforced leave before beginning the MOS course, then the back-to-back MOS class will be scheduled to start after the completion of these activities.

The fourth principle is used to compute the *TBSSIZE(i)* values. TBS classes with no back-to-back MOS classes will always assign exactly c officers to the MOS. The remaining officers to fill the MOS quota will be equally distributed among TBS classes with back-to-back MOS classes. An example will help to clarify how this is done. Assume there are eight TBS classes and each TBS class must assign at least one officer to the MOS. Also, the MOS course has four classes which are scheduled so that each MOS class begins when an alternate TBS class finishes, i.e. every second TBS class has a back-to-back MOS class. There are forty officers to be assigned into the MOS from TBS classes. In this example, the heuristic will assign nine officers from the four back-to-back TBS classes and one officer each from the other TBS classes. Evidently, there is no way to improve this assignment without violating at least one of the constraints, although conceivably, there are other ways to achieve the same result.

The final principle aims to equalize the *TBSSIZE(i)* values by always attempting to schedule a back-to-back MOS class with the TBS class that has the smallest *TBSSIZE*, and also avoiding the scheduling of a back-to-back class to the TBS class with the largest *TBSSIZE*. This is because of the class size restriction that every TBS class must observe. Evidently, if every *TBSSIZE(i)* value falls within the TBS class size limits after all MOS courses have been scheduled, it will mean that apart from the minimum assignments, TBS classes can assign the rest of their officers into back-to-back MOS classes. Assuming the problem is feasible, then equalizing the *TBSSIZE(i)* values will keep them within the permissible TBS class size range.

4. Data Inputs

Data for the model is stored in a single input file which is divided into four sections. The contents of each section are listed below:

(a) *OCS data*

- OCS class start dates
- Number of Air OCC officers per OCS class

(b) *TBS data*

- TBS class start dates
- TBS class end dates
- Warrant officer TBS class number

(c) *MOS course data*

- Duration of MOS course
- Minimum assignment into MOS from each TBS class
- MOS output required from TBS graduates

- MOS output required from non-TBS sources
- On-the-Job training period required before MOS course
- Enforced leave period before MOS course
- End date of the last MOS class scheduled in the previous FY
- Indicator variable for preselecting MOS course schedule
- Start dates of MOS classes (only for non-USMC controlled MOS courses)
- Seat allocations for MOS classes (only for non-USMC controlled MOS courses)

(d) *Preselected course data*

- Number of classes in preselected MOS course schedule
- Preselected MOS course schedule

A sample input file is shown in Appendix B.

5. Course Scheduling Algorithm

The algorithm consists of several subroutines and a main program that controls the order of execution.

a. Main Program

This forms the heart of the algorithm by controlling its entire operation. A flowchart of the program is shown in Figure 2.2. At program initialization, the *subroutine INPUT* reads in data from the input file. Also, it calls the *subroutine INIT* to compute the starting values for *TBSSIZE*, i.e., before USMC-controlled MOS assignments are considered. Before an iteration is started, a check is made to see if classes for all MOS's have been scheduled. If so, then *subroutine OUTPUT* is called upon to generate the results; otherwise, the next MOS is called. Each iteration produces the course schedule for a USMC controlled MOS school. The order for scheduling MOS courses is defined by a *Priority List* which ranks the MOS's in descending order according to the value of their minimum assignment from each TBS class. An iteration first goes through a decision point to check if the MOS course schedule has been preselected by the user. If so, *subroutine PRESEL* is called; otherwise *subroutine RANK* is called to rank the TBS classes according to their TBSSIZE values. Then it proceeds to the next decision point which checks if the MOS allows concurrent classes. If so, *subroutine CONCUR* is called; if not, it calls *subroutine CONSEC*. In the final step, it calls *subroutine UPDATE* to compute the latest *TBSSIZE* values. The program then returns to check if all MOS courses have been scheduled.

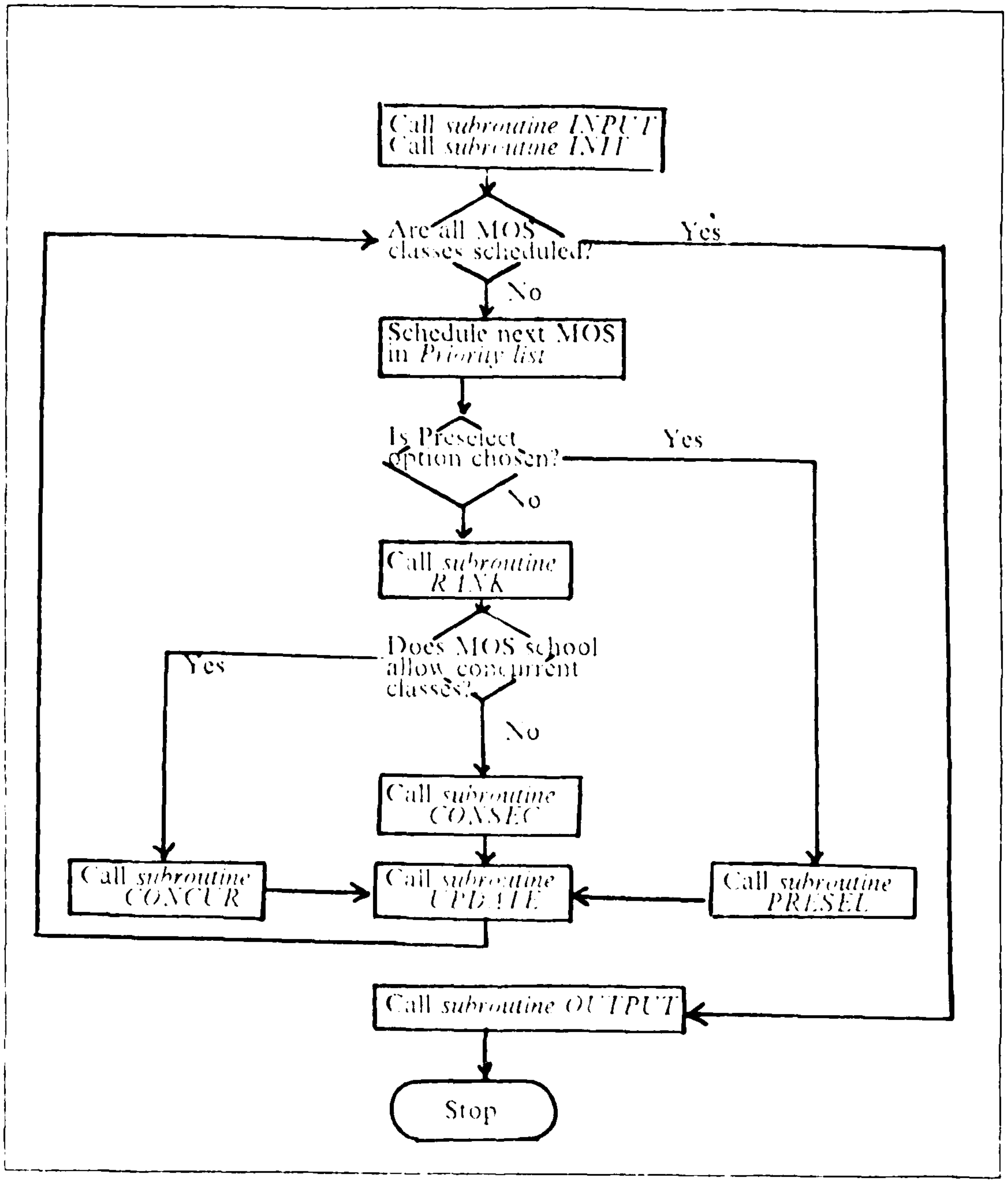


Figure 2.2 Flowchart for Main Program.

b. Subroutines

There are fourteen subroutines:

- *Subroutine INPUT* reads in data from the first three sections of the input file. It also computes the number of classes to be scheduled for each USMC controlled MOS course by following *heuristic principle one*.

- *Subroutine INIT* computes the values for *TBSSIZE* formed by Air officers and Ground officers attending non-USMC MOS courses. The updating of *TBSSIZE* values for Air officers is straight forward. OCC Air officers will be counted under the TBS class which carries zero waiting time for assignments. Non-OCC Air officers are identified by the TBS class they have been assigned to by HQMC. The accounting for Ground officers is more complex. *Subroutine TBSPL* is called to provide a list of non-USMC course places that will be filled by TBS assignments. The assignments are then broken down by TBS classes so as to compute *TBSSIZE.*

- *Subroutine TBSPL* identifies the non-USMC MOS course places to be occupied by TBS assignments. A separate iteration is performed for each non-USMC controlled MOS course. An iteration has the following steps:

  (a) *Step 1*
  Assign the variable $wait_j$ as the delay if an assignment was made into MOS class j from the TBS class carrying the smallest delay.

  (b) *Step 2*
  Sort the MOS classes in ascending order according to $wait_j$.

  (c) *Step 3*
  Fill each MOS class (in the sorted order) with TBS assignments until the MOS quota for TBS assignments is reached.

- *Subroutine PRESEL* assigns the preselected MOS course schedule read from the input file to the specified MOS. Further, it schedules an additional MOS class to start after the ending date of the last TBS class if the preselected course schedule does not allow assignments to be made from all TBS classes.

- *Subroutine RANK* sorts the TBS classes according to *TBSSIZE* values. Specifically, it produces an array such that the ith element represents the class number of the TBS class with the ith smallest *TBSSIZE* value.

- *Subroutine CONCUR* develops the schedule for a USMC controlled MOS school with concurrent classes. It follows *heuristic principle five* in trying to achieve *TBSSIZE* values that lie between 150 and 250 after all MOS's are scheduled. Each MOS course schedule is developed in such a way as to equalize the *TBSSIZE* values as much as possible. This may require several iterations of the subroutine.

  The starting iteration will schedule the first MOS class to be back-to-back with the TBS class having the smallest *TBSSIZE.* Remaining classes are scheduled by calling *subroutine REMCUR.* Then *subroutine FEAS1* is called to detect and correct infeasibilities in the initial course schedule. If the corrected course schedule has a back-to-back MOS class to the TBS with the largest *TBSSIZE* (label this class *TBS(largest)*), another iteration is performed to produce a new schedule. In this iteration, the first MOS class is scheduled to be back-to-back with the TBS class having the next smallest *TBSSIZE.* The same process as before will be repeated until a schedule with no back-to-back class to *TBS(largest)* is created. The steps are:

  (a) *Step 1*

Let $j = 1$

(b) *Step 2*
Let *index* be the class number of the TBS class with the *jth* smallest *TBSSIZE*.

(c) *Step 3*
Let $class(1) = TBSEND(index) + OJT + LV$

(d) *Step 4*
Call *subroutine REMCUR* to schedule the remaining classes.

(e) *Step 5*
Call *subroutine FEAS1* for feasibility corrections on the initial course schedule.

(f) *Step 6*
If back-to-back MOS class is scheduled for the TBS class with largest *TBSSIZE*
THEN
let $j = j + 1$
Goto step 2
ELSE
Stop

- *Subroutine REMCUR* schedules the remaining classes for a USMC controlled MOS course with concurrent classes after *subroutine CONCUR* has scheduled the first class. A variable LAG_j is assigned to each unscheduled MOS class j. LAG_j denotes the number of non back-to-back TBS classes between MOS classes j and $j+1$ and is computed following *heuristic principle two* by spreading MOS classes "evenly" among the TBS classes. In cases where the number of TBS classes is not divisible by the number of MOS classes, an arbitrary tie breaking rule is used to derive LAG_j. The exact date to fix a MOS class is determined using *heuristic principle three* by always scheduling the MOS class so that it begins when its back-to-back TBS class ends. The steps are:

(a) *Step 1*
Let $last = index$
where the label *index* has been defined previously in *subroutine CONCUR*.
Assign the variable LAG_j to MOS class j for $j = 2,..n$

(b) *Step 2*
Let $j = 2$
where j is the MOS class index.

(c) *Step 3*
If $j \leq n$
THEN
LET $class(j) = TBSEND(last + LAG_j) + OJT + LV$
$last = last + LAG_j$
$j = j + 1$
Repeat this step
ELSE

stop

- *Subroutine FEAS1* checks for feasibility of the schedule produced for a USMC MOS course with concurrent classes. Appropriate adjustments are made in cases where infeasibilities are detected. Before these checks are carried out, the MOS class start dates have to be adjusted for them to observe the same phase cycle as the TBS class schedule. This step involves the use of modular arithmetic. If p is a real number, d is a positive integer and x the remainder when p is divided by d, then x equals p *modulo* d. The adjustment of MOS class start dates is achieved by performing a modulo 52 operation on each date, after which the classes are sorted according to their new start dates. There are two checks performed by the subroutine. First, it checks if $class(1)$ is scheduled to start after $TBSEND(1) + OJT + LV$. Next, it checks if $class(n)$ is scheduled after $TBSEND(8) + OJT + LV$. The steps are:

  (a) *Step 1*
  Let $class(i) = class(i)$ *Mod 52* for $i = 1,...n$
  Sort classes according to start dates

  (b) *Step 2*
  If $class(1) \leq TBSEND(1) + OJT + LV$
  THEN
  Let $class(1) = class(1) + 52$
  Sort classes according to start dates

  (c) *Step 3*
  If $class(n) \geq TBSEND(8) + OJT + LV$
  THEN
  Let $m = n$
  ELSE
  Let $m = n + 1$
  $class(m) = class(1) + 52$

- *Subroutine CONSEC* schedules the classes for a USMC controlled MOS course with non-overlapping classes. It has an almost identical structure to *subroutine CONCUR*. An additional step is included in *subroutine CONSEC* to check if the course schedule can be finished in 52 weeks. If not, *subroutine PUSH* is called to perform the adjustments for fitting the course schedule into 52 weeks. The subroutine has seven steps:

  (a) *Step 1*
  Let $j = 1$

  (b) *Step 2*
  Label $TBSEND(index)$ to be the end date of the TBS class with the *jth* smallest *TBSSIZE*.

  (c) *Step 3*
  Let $class(1) = TBSEND(index) + OJT + LV$.

  (d) *Step 4*
  Call *subroutine REMSEC* to schedule the remaining classes.

(e) *Step 5*

If the classes scheduled cannot be completed within 52 weeks

THEN

Call *subroutine PUSH* to develop a schedule which can be completed within 52 weeks

(f) *Step 6*

Call *subroutine FEAS2*

(g) *Step 7*

If a back-to-back MOS class is scheduled for the TBS class with largest *TBSSIZE*

THEN

let $j = j + 1$

Goto step 2

ELSE

Stop

- *Subroutine REMSEC* has a function similar to *REMCUR*, differing only because it schedules classes for a MOS course which does not allow class overlaps. An extra step is incorporated in *subroutine REMSEC* to check for and correct cases where overlapping classes have been scheduled. The steps for *subroutine REMSEC* are:

(a) *Step 1*

Let $last = index$

where the label *index* has been defined previously in *subroutine CONSEC*. Assign the variable LAG_j to MOS class j for $j = 2...n$

(b) *Step 2*

Let $j = 2$

where j is the MOS class index.

(c) *Step 3*

If $j \leq n$

THEN

LET $trial = TBSEND(last + LAG_j) + OJT + LV$

Goto step 4

ELSE

stop

(d) *Step 4*

If $trial \geq class(j-1) + d$

THEN $class(j) = trial$

$last = last + LAG_j$

$j = j + 1$

Goto step 3

ELSE

$class(j) = TBSEND(first) + OJT + LV$

where *first* is the label assigned to the first TBS class which is allowed to be a back-to-back class with MOS class j
$last = first$
$j = j + 1$
Goto step 3

- *Subroutine PUSH* is called whenever a MOS course schedule cannot be completed within 52 weeks. The subroutine produces an initial schedule where each class begins as soon as the preceeding class ends. Clearly, this schedule is feasible (i.e. it can be completed in 52 weeks) although steps could be taken to improve it. This can be done as follows. The first MOS class will always begin when its back-to-back TBS class ends, since this is the way it was scheduled by *subroutine CONSEC*. Starting with the second MOS class, each class is checked to see if it begins when its back-to-back TBS class ends. If not, and if it is feasible to do so, each class taken one at a time, will be slided forward so as to begin at the end date of the next feasible TBS class. This sliding operation terminates when either all MOS classes have been appropriately rescheduled or when there is not enough *slack* time left to continue with the rescheduling. The steps are:

(a) *Step 1*
Set $slack = 52 - n*d$
where d is the duration of a MOS course

(b) *Step 2*
Let $class(1) = TBSEND(index) + OJT + LV$
where $TBSEND(index)$ has been defined previously in *subroutine CONSEC*.

(c) *Step 3*
Assign $class(j) = class(j-1) + d$
for $j = 2,...n$

(d) *Step 4*
Let $j = 2$

(e) *Step 5*
If $j \leq n$
THEN
Goto step 6
ELSE
stop

(f) *Step 6*
If $slack \geq 0$
THEN
Goto step 7
ELSE
stop

(g) *Step 7*

Assign the label *first* to the first TBS class which is allowed to be a back-to-back class with MOS class j
If $class(j) + slack \geq TBSEND(first) + OJT + LV$
THEN
$slack = slack\text{-}(TBSEND(first) + OJT + LV\text{-}class(j\text{-}1))$
$class(j) = TBSEND(first) + OJT + LV$
Let $class(k) = class(k\text{-}1) + d$ for $k = j + 1,..n$
$j = j + 1$
Goto step 5
ELSE
Goto step 5

- *Subroutine FEAS2* performs the feasibility checks for a course schedule that has been produced for a USMC controlled MOS school with non-overlapping classes. It has the features of the *subroutine FEAS1* and in addition, has a step to correct for cases where the first of the current FY's MOS classes overlaps with the last MOS class scheduled in the previous FY. The subroutine has the following steps:

  (a) *Step 1*
  Let $class(i) = class(i)\ Mod\ 52$ for $i = 1,...n$
  Sort classes according to start dates

  (b) *Step 2*
  If $class(1) \leq TBSEND(1) + OJT + LV$
  THEN
  Let $class(1) = class(1) + 52$
  Sort classes according to start dates

  (c) *Step 3*
  If $class(1) \leq PREV$
  where $PREV$ is the end date of the last MOS class scheduled in the previous FY
  THEN
  Let $class(1) = class(1) + 52$
  Sort classes according to start dates

  (d) *Step 4*
  If $class(n) \geq TBSEND(8) + OJT + LV$
  THEN
  Let $m = n$
  ELSE
  Let $m = n + 1$
  $class(m) = class(1) + 52$

- *Subroutine UPDATE* keeps track of the variable *TBSSIZE*. After classes for a USMC-controlled MOS school are scheduled, the subroutine is called to compute the assignments from each TBS class into the MOS; the results are then used to update the *TBSSIZE* values.

The method for computation focuses on the difference between TBS classes with back-to-back classes and those without. Assignments from TBS classes with back-to-back MOS classes will have zero or at most small delays, and those from classes with no back-to-back MOS classes will experience considerably longer delays.

Heuristic principle four is used to compute the *TBSSIZE* values as follows. TBS classes with no back-to-back MOS classes would assign exactly c officers to the MOS. This is the minimum assignment required for problem feasibility. The update for TBS classes with back-to-back MOS classes has more steps. First, non back-to-back TBS classes are separated into three groups and the numbers in each group counted: TBS classes ending after $class(n)$ (number in this group is labelled *AFTER*), TBS classes ending before $class(1)$ (number in this group is labelled *BEFORE*), and TBS classes ending between $class(j)$ and $class(j-1)$ where $j=2,..n$ (number of non back-to-back TBS classes between $class(j)$ and $class(j-1)$ is labelled $t(j)$) . Next, a check is made to see if a given TBS class is back-to-back with the first MOS class. If so, then the number of assignments from the TBS class is given by $q_i n$ - c*($AFTER + BEFORE$). If not, using the notation that TBS class i is back-to-back with MOS class j, the number of assignments from TBS class i is given by $q_i n$ - c*$t(j)$.The steps in the subroutine are:

(a) *Step 1*
Let *AFTER* be the number of TBS classes ending after $class(n)$

(b) *Step 2*
Let *BEFORE* be the number of TBS classes ending before $class(1)$

(c) *Step 3*
Let $t(j)$ be the number of non back-to-back TBS classes between $class(j)$ and $class(j-1)$ for $j=2,...n$

(d) *Step 4*
Let $i=1$

(e) *Step 5*
If $i \leq 8$
where i is the TBS class index
THEN
Goto step 6
ELSE
stop

(f) *Step 6*
If TBS class i has a back-to-back MOS class
THEN
Assign *star* as the label for the back-to-back MOS class to TBS class i
Goto step 7
ELSE
Let $TBSSIZE(i) = TBSSIZE(i) + c$
$i = i + 1$
Goto step 5

(g) *Step 7*

If MOS class *star* is the first MOS class (i.e. *star* equals one)
THEN
Let $TBSSIZE(i) = TBSSIZE(i) + q_in\text{-}c*(AFTER + BEFORE)$
$i = i + 1$
Goto step 5
ELSE
Let $TBSSIZE(i) = TBSSIZE(i) + q_in\text{-}c*t(star)$
$i = i + 1$
Goto step 5

- *Subroutine OUTPUT* produces the two output files. The first file formats the information for easy human intepretation and the second is designed for providing input to the officer assignment program. There is an example of the second output file in Appendix C corresponding to the input of Appendix B.

C. OFFICER ASSIGNMENT MODEL

The next step after the course schedule is developed is to solve the officer assignment problem. The Linear Programming (LP) formulation of the problem is explained as follows:

Indices:

h	OCS class numbers
i	TBS class numbers
j	MOS class numbers
k	Types of MOS courses

Sets:

M1	USMC controlled MOS courses
M2	MOS courses where unrestricted and warrant officers are assigned to the same MOS classes

Parameters (given data) :

(a) *Quotas*

Q_k	Output quota for MOS k
ST_k	Total number of officers from last year's TBS classes assigned to current FY classes for MOS k
VT_k	Total number of voluntary lateral move officers assigned to MOS k
DT_k	Total number of directed lateral move officers assigned to MOS k
IT_k	Total number of Intended MOS officers assigned to MOS k
FT_k	Total number of grounded Air officers assigned to MOS k

WO_k	Number of warrant officers assigned to MOS k
AO_h	Number of OCC Air officers graduating from OCS class h
GO_h	Number of OCC Ground officers graduating from OCS class h
AT_i	Number of non-OCC Air officers assigned to enter TBS class i

(b) *Training school restrictions*

m_k	Minimum assignment into MOS k from each TBS class
ml_k	Minimum class size for MOS course k
mu_k	Maximum class size for MOS course k

(c) *Waiting times*

$W1_{hi}$	Delay for assignment from OCS class h to TBS class i
$W2_{ikj}$	Delay for assignment from TBS class i to class j of MOS course k
$W3_{kj}$	Delay for officers from last year's TBS classes assigned to class j of MOS course k

(d) *Other*

w	Warrant officer TBS class number

Decision Variables:

(a) *Assignments into TBS*

AX_{hi}	Number of OCC Air officers assigned from OCS class h to TBS class i
GX_{hi}	Number of OCC Ground officers assigned from OCS class h to TBS class i
GT_i	Number of non-OCC Ground officers assigned to enter TBS class i

(b) *Assignments from TBS to MOS classes*

Y_{ikj}	Number of officers assigned from TBS class i to class j of MOS k

(c) *Assignments from non-TBS sources to MOS classes*

V_{kj}	Number of voluntary lateral move officers assigned to class j of MOS course k
D_{kj}	Number of directed lateral move officers assigned to class j of MOS course k
F_{kj}	Number of grounded Air officers assigned to class j of MOS course k
I_{kj}	Number of Intended MOS officers assigned to class j of MOS course k
S_{kj}	Number of last year's TBS graduates assigned to class j of MOS k

(d) *Class sizes*

TBS_i Size of TBS class i

MOS_{kj} Size of class j of MOS course k

The objective is to minimize the total course waiting time for all officers. By definition, officers assigned to MOS classes from non-TBS sources[4] will experience zero delay. The objective function can be stated thus:

$$\text{Minimize} \quad \sum_h \sum_i{}^{5} W1_{hi}*(AX_{hi}+GX_{hi}) + \sum_i \sum_k \sum_j W2_{ikj}*Y_{ikj} + \sum_k \sum_j W3_{kj}*S_{kj}$$

The model has fourteen sets of constraints. Constraints (1) ensure that the number of OCC Air officers assigned from each OCS class equals the numbers graduating from the class. Constraints (2) enforce the same restriction for OCC Ground officers. The equations for these two constraints are:

$$\sum_i AX_{hi} = AO_h \qquad \textit{Constraints (1)} \qquad \textit{for all } h$$
$$\sum_i GX_{hi} = AO_h \qquad \textit{Constraints (2)} \qquad \textit{for all } h$$

Constraints (3) and (4) combine to preserve the flow of officers through each unrestricted officer TBS class. Constraints (3) sums the inflow of officers into a unrestricted officer TBS class *i*, and equates it to variable TBS_i:

$$\sum_h (AX_{hi} + GX_{hi}) + GT_i + AT_i = TBS_i \qquad \textit{Constraints(3)} \qquad \textit{for all } i \neq w$$

Constraints (4) sum the outflow of officers from the unrestricted officer TBS class *i* and equates it also to TBS_i. Since the assignment of Air officers into MOS classes is not considered by the proposed model, the outflow for these officers is simply equated with their inflow. Constraints (4) appear as:

$$TBS_i = \sum_k \sum_j Y_{ikj} + \sum_h AX_{hi} + AT_i \qquad \textit{Constraints(4)} \qquad \textit{for all } i \neq w$$

[4]Recall that officers forming these groups are those who are laterally moved (either voluntarily or directed by HQMC), officers under the Intended MOS scheme and grounded Air officers.

[5]For notational brevity, in this section, whenever the symbol $\sum$ is used, it is assumed that the summation will be performed over feasible values of the index set. In the model's implementation, the restrictions are enforced using the GAMS "dollar" (or "such that") operator.

Constraints (5) and (6) relate to the warrant officer TBS class. Constraint (5) equates the variable TBS_w to the total number of warrant officers from MOS's belonging to set M2:

$$TBS_w = \sum_{k \in M2} WO_k \qquad \textit{Constraint(5)}$$

Constraints (6) have an equation for each MOS belonging to set M2. For the given MOS, it sums the assignments from the warrant officer TBS class to all feasible classes of the MOS, and equates it to the total number of warrant officers to be channeled into the MOS, that is:

$$\sum_j Y_{wkj} = WO_k \qquad \textit{Constraints(6)} \quad \textit{for } k \in M2$$

Constraints (7) impose the minimum MOS assignment restriction for each unrestricted officer TBS class and into every MOS as follows:

$$\sum_j Y_{ikj} \geq m_k \qquad \textit{Constraints(7)} \quad \textit{for } i \neq w \textit{ and for all } k$$

Constraints (8) sums the number of officers assigned to class *j* of MOS course *k* and equates it to variable MOS_{kj}:

$$\sum_i Y_{ikj} + S_{kj} + V_{kj} + D_{kj} + F_{kj} + I_{kj} = MOS_{kj} \qquad \textit{Constraints(8)} \quad \textit{for all } k \textit{ and } j$$

Constraints (9) enforce the requirement for each MOS course to meet its MOS output quota:

$$\sum_j MOS_{kj} = Q_k \qquad \textit{Constraints(9)} \quad \textit{for all } k$$

Finally, constraints (10) to (14) balance the assignments from each non-TBS source with the total number of officers available from that source. Equations for the five constraints are:

$$\sum_k \sum_j S_{kj} = ST_k \qquad \textit{Constraints(10)} \quad \textit{for all } k$$
$$\sum_k \sum_j V_{kj} = VT_k \qquad \textit{Constraints(11)} \quad \textit{for all } k$$
$$\sum_k \sum_j D_{kj} = DT_k \qquad \textit{Constraints(12)} \quad \textit{for all } k$$
$$\sum_k \sum_j I_{kj} = IT_k \qquad \textit{Constraints(13)} \quad \textit{for all } k$$
$$\sum_k \sum_j F_{kj} = FT_k \qquad \textit{Constraints(14)} \quad \textit{for all } k$$

There are two sets of variable bounds in addition to the non negativity restriction for all variables. These are :

$$150 \le TBS_i \le 250 \qquad \textit{for all } i \ne w$$

$$ml_k \le MOS_{kj} \le mu_k \qquad \textit{for all } j \textit{ and for } k \in MI$$

Using this formulation, the problem can be solved easily with a LP solver. The details on how this is done will be covered in the next chapter.

III. IMPLEMENTATION AND COMPUTATIONAL RESULTS

This chapter explains the implementation procedures for the algorithms used by the heuristic model. It also describes the data used for computational test runs, before presenting the results of these tests.

A. IMPLEMENTATION

Implementation into computer programs are discussed separately for the course scheduling model and the officer assignment model.

1. Course Scheduling Model

A *Fortran 77* implementation is used to solve the course scheduling problem on either an IBM 3033AP mainframe or an IBM personal computer(XT or AT). The two versions require different compilers--the mainframe version has been successfully tested with a VS FORTRAN compiler and the PC version with a Ryan Mcfarland RM FORTRAN compiler. The code for the FORTRAN program is kept on electronic media which can be obtained by contacting either the thesis advisors or the Operations Research Department at the Naval Postgraduate School. Data required to run the program is stored in a separate input file. The entries required for this file was described in Chapter Two.

Apart from developing course schedules, the course scheduling program serves a secondary purpose as a management tool for examining policy tradeoffs. The program produces an output table called "Waiting Time Report" (see section 3(C)) which shows the waiting times for officers assigned from TBS classes into follow-on MOS classes. If a MOS course has a back-to-back class scheduled with TBS class i, then the waiting time for officers assigned from TBS class i to the MOS will be very small and probably zero. However, if there is no back-to-back MOS class for TBS class i, the waiting time for officers assigned into the MOS will be considerably longer. HQMC has a policy requiring each unrestricted officer TBS class to assign a prescribed minimum number of officers to certain designated MOS's. To adhere to this policy for any given MOS would imply that some officers entering the MOS from TBS classes must be assigned into non back-to-back MOS classes. This is because not every TBS class will have a back-to-back MOS class. Thus, the table displays to the user the "cost" of imposing such minimum assignment restrictions. To assist the user in

exercising his prerogative, an option is created in the officer assignment program to allow relaxation of this restriction.

2. Officer Assignment Model

The *GAMS* Modeling system [Ref. 2,3,4,5,6] is used to solve the officer assignment problem. The problem can be solved on mainframe or personal computers. A listing of the officer assignment program is given in Appendix D. The data inputs to run the program are divided between those supplied by the course scheduling program and those coming directly from HQMC. Inputs from the two groups are listed below under separate headings. Where applicable, the notation used by the model formulation in Chapter Two will be shown in parentheses after the name of the input:

(a) *Inputs from course scheduling program*

- Start dates for all MOS classes
- Number of classes to be conducted for each MOS course

(b) *Inputs directly from HQMC*

- Annual MOS output quota (Q_k)
- Output from previous FY's TBS classes by MOS (ST_k)
- Number of Warrant officers assigned to "mixed" MOS's by MOS (WO_k)
- Warrant officer TBS class number (w)
- Number of Air officers rechannelled to Ground MOS's by MOS (FT_k)
- Number of voluntary lateral movements by MOS (VT_k)
- Number of directed lateral movements by MOS (DT_k)
- Number of officers under the Intended MOS scheme by MOS (IT_k)
- Number of OCC Ground officers entering each OCS class (GO_h ÷ OCS success rate)
- Number of OCC Air officers entering each OCS class (AO_h ÷ OCS success rate)
- Number of non-OCC Air officers entering each TBS class (AT_i)
- Course seat allocation at non-USMC MOS schools by individual class
- Upper capacity for MOS class by MOS (mu_k)
- Lower capacity for MOS class by MOS (ml_k)
- Starting week of each OCS class
- Starting week of each TBS class
- Ending week of each TBS class

As mentioned in the preceeding sub-section, the program has an option to relax the restriction for each MOS to be assigned a prescribed minimum number of

officers from every unrestricted officer TBS class. There are two ways to do this. depending on how the restriction is to be relaxed. The first way will relax the restriction whenever an assignment from TBS into a follow-on MOS class has to encounter a delay greater than some level chosen by the user. The second enables the user to select the MOS as well as the specific TBS classes for which the restriction is to be relaxed. Detailed instructions on how to use this option is written in the GAMS program and is also described in the user instructions contained in Appendix G.

B. TEST DATA

The test data was extracted from documents supplied by the Manpower Assignment Section in HQMC. All of the information supplied represent real data for FY 88 except for the dates of non-USMC controlled MOS classes which are available only for classes starting between January 87 and December 87. To develop a test solution for FY 88, it was assumed that corresponding non-USMC classes would also be available in FY 88. As part of the data extraction process, the dates of classes have to be converted into the same time scale as that used by the model, i.e., weeks counted from an arbitrary origin. Appendix F is a chart showing the time conversion.

C. COMPUTATIONAL RESULTS

The Fortran 77 program for the course scheduling algorithm was used to develop a course schedule for fifteen USMC MOS schools. Each school has an average of three classes to schedule. The running times for the program are shown in Table 1:

TABLE 1
COMPUTATION TIMES FOR COURSE SCHEDULING MODEL

Device	Compilation time	Execution time
IBM 3033AP Mainframe	3.7 secs	0.8 secs
IBM PC-AT at 8 mhz *(80287 math coprocessor)*	4.2 mins	10.2 secs

Output from the Fortran 77 program is sent to two different files. The first file in the PC implementation is called SCHED.GMS and it formats the output for entry into the GAMS program. The second file, USER.OUT, has the output formatted for human intepretation. Both files are commented so as to be self-documenting. Appendix C shows a sample of file SCHED.GMS. The output used in file USER.OUT is shown below to explain the results.

This file has three sections. Section one is the TBS schedule report which appears as follows:

***** TBS SCHEDULE REPORT *****

CLASS	FINISH DATE	UNRESTRICTED OFFICERS	WARRANT OFFICERS	TOTAL ENROLMENT
TBS1	39	164		164
TBS2	44	179		179
TBS3	52	180		180
TBS4	57	134		134
TBS5	62	225		225
TBS6	74	133		133
TBS7	65		37	37
TBS8	79	135		135
TBS9	86	150		150

The total enrolment for each TBS class corresponds to its *TBSSIZE* value after all MOS courses are scheduled. Since there are no class size restrictions on the warrant officer TBS class, only the *TBSSIZE* values for unrestricted officer TBS classes will be of interest. The results show *TBSSIZE* values for these classes ranging between 133 to 225. The lower value indicates that apart from officers who are assigned because of the minimum assignment restriction, some TBS class(es) may have to assign additional officers to enter non back-to-back MOS classes. This would depend on whether there exists a feasible solution which allows TBS classes to assign all officers (apart from those assigned because of the minimum assignment restriction) into back-to-back MOS classes. If this solution exists, an alternative method of computation would have yielded *TBSSIZE* values all within the TBS class size limits. The best way to verify is to execute and check the results of the officer assignment program using the course schedules that have been produced.

The next section shows the MOS schedule report:

***** MOS SCHEDULE REPORT *****

NAME	NUMBER OF CLASSES	MAXIMUM DELAY(WKS)	START DATES C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
ARTY	8	9	46	49	54	59	65	73	85	92		
TANK	5	13	48	62	77	83	92					
ADA	10	11	42	50	56	61	65	71	75	80	84	88
AMO	7	12	55	58	66	74	78	81	98			
PAO	4	11	46	56	65	75						
MPO	8	8	49	53	59	63	66	72	76	81		
INFAN	8	9	39	44	52	57	62	74	79	86		
COMM	3	17	54	76	90							

SUP	5	14	41	54	66	81	93
AIRSUP	5	14	41	54	64	81	93
MT	4	22	41	54	81	93	
LOGS	3	22	54	81	106		
ENG	5	10	46	59	71	83	98
FIN	2	24	59	88			
ADC	4	12	50	63	76	89	
AVNSP	4	14	56	74	91	108	
ADP	3	22	46	76	98		
ATC	3	22	54	81	106		
AMPH	2	24	59	88			
ADJ	3	22	46	76	98		
INT	4	12	46	59	76	88	

The figures under the column NUMBER OF CLASSES are derived from different sources depending on the MOS course. Those for non-USMC controlled MOS courses are the same as given in the input file. For USMC controlled MOS courses, the figure is computed by the heuristic method described in Chapter Two. The next column lists the maximum delay (in weeks) for a TBS assignment entering a follow-on MOS class. For example, the maximum delay for an Artillery officer is nine weeks because this is the longest period he has to wait to enter a follow-on Artillery MOS class from TBS. Finally, the start dates for all MOS classes are shown.

The last section shows the delay for assignments into follow-on MOS classes from each of the nine TBS classes under the heading WAITING TIME REPORT:

***** WAITING TIME REPORT *****
(DELAY MEASURED IN WEEKS)

	TBS1	TBS2	TBS3	TBS4	TBS5	TBS6	TBS7	TBS8	TBS9
ARTY	5	0	0	0	1	9	6	4	4
TANK	7	2	8	3	13	1	10	2	4
ADA	1	4	2	2	1	4	4	3	0
AMO	4	2	2	5	0	12	1	7	0
PAO	5	0	2	6	1	0	8	0	0
MPO	8	3	5	0	2	0	5	0	0
INFAN	0	0	0	0	0	0	9	0	0
COMM	13	8	0	17	12	0	9	9	2
SUP	0	8	0	7	2	5	14	0	5
AIRSUP	0	8	0	5	0	5	14	0	5
MT	0	8	0	22	17	5	14	0	5
LOGS	13	8	0	22	17	5	14	0	18
ENG	5	0	5	0	7	7	4	2	10
FIN	18	13	5	0	24	12	21	7	0
ADC	9	4	9	4	12	0	9	8	1
AVNSP	5	0	10	5	0	5	14	0	10
ADP	5	0	22	17	12	0	9	17	10
ATC	13	8	0	22	17	5	14	0	18
AMPH	18	13	5	0	24	12	21	7	0
ADJ	5	0	22	17	12	0	9	17	10
INT	5	0	5	0	12	0	9	7	0

As an example, from the above printout, the delay for an assignment from TBS class one to a follow-on class for the ARTY (Artillery) MOS course is five weeks.

The running times using different devices for the GAMS officer assignment program are shown in Table 2:

TABLE 2

COMPUTATION TIMES FOR OFFICER ASSIGNMENT MODEL

Device	Compilation time	Execution time
IBM 3033AP Mainframe	10.7 secs	15.9 secs
IBM PC-AT at 8 mhz *(80287 math coprocessor)*	2.1 mins	5.1 mins
IBM PC-XT at 4.77 mhz *(8087 math coprocessor)*	5.1 mins	12.0 mins

Several runs of the GAMS program were executed, each based upon a different relaxation of the minimum MOS assignment restriction from TBS classes. The five sets of conditions used to generate these runs are:

(a) At least one officer to be assigned from each TBS class to every MOS except for MOS's where the total assignments from TBS classes is less than eight. For the latter MOS's, the minimum assignment is zero.

(b) At least x_k officers to be assigned from each TBS class to MOS k where x_k is five percent of the total assignments from TBS classes to MOS k. As before, MOS's with a total of less than eight officers to be assigned from TBS classes are not required to observe the minimum assignment restriction.

(c) As in condition (b) except that for a given MOS, the restriction is relaxed if the delay for assignments from followed-on TBS classes is greater than four weeks.

(d) As in condition (b) except that for a given MOS, the restriction is relaxed if the delay for assignments from followed-on TBS classes is greater than eight weeks.

(e) No restriction is imposed on minimum MOS assignment from TBS classes.

The output provided by the solver is, quoting from the GAMS documentation, "rich in detail". For brevity, only the variable listing of the solution report is presented in Appendix E. A summary of the results from the five test runs is shown in Table 3.

TABLE 3

SUMMARY OF TEST RESULTS

Restriction on minimum assignment	Total waiting time in man weeks	Average waiting time in man weeks
Condition (a)	1745	1.8
Condition (b)	2142	2.3
Condition (c)	1033	1.1
Condition (d)	1361	1.4
Condition (e)	1033	1.1

Total and average waiting times as indicated above are measured in man weeks. Average waiting time is computed by dividing total waiting time by the number of Ground officers attending TBS classes, i.e., a total of 950 officers for FY 88. The FY 88 solution will closely follow the test run results since real data except for non-USMC MOS course allocations was used for the test problem. These results can also be gauged against those from previous years. Although HQMC does not maintain a complete record of past waiting times, it is possible to derive an estimate from available documents. An internal document by HQMC on the subject shows the average waiting time per man in the last two years to be at least five weeks. Against this figure, the results from the proposed model show up favorably; even under the most restrictive condition, a significantly lower average waiting time is obtained than in the past.

IV. CONCLUSION

This thesis proposes a heuristic model to solve the course scheduling and assignment problem for officers undergoing initial skill training in the United States Marine Corps. Although the approach does not guarantee an optimal solution, in this problem, heuristics make it computationally feasible to develop a model that could handle all the constraints specified by the user. An unsuccessful attempt was made to solve a reduced scale formulation using mixed integer programming on the IBM 3090 mainframe computer at the World Bank. Other optimization procedures have been explored but none implemented because of the complexity of the problem.

The proposed model solves the problem by dividing it into two sub-problems. The first sub-problem is to develop the course schedules for designated officer schools under the control of the USMC. The second sub-problem solves the officer class assignments. The algorithms used for problem solving have been implemented into programs that run on both mainframe and personal computers. A Fortran 77 implementation is used for the course scheduling problem. The program designed for execution on the IBM 3033AP mainframe system uses a VS FORTRAN compiler. The PC version requires a Ryan Mcfarland RM/FORTRAN compiler. The officer assignment problem is solved with the GAMS Modeling system which has a common version for mainframe and personal computers.

The computer programs were checked by using them to solve a test problem. Data for the test problem was extracted from information provided by HQMC. Apart from the start dates for non-USMC controlled MOS courses, which has not been received from conducting agencies, the information represents actual data used by HQMC for FY 88 planning. Five different test runs were executed. Each run carried a different assumption about the policy set by HQMC for mandatory MOS assignments from TBS classes. The test run results were compared to actual results over the last two years using average waiting time per man as the criteria. The answer by using the proposed model ranges from 1.1 weeks when no mandatory MOS assignment restriction is imposed, to 2.3 weeks when the general rule requires an unrestricted officer TBS class to assign into each MOS at least five per cent of the MOS's output quota of officers from TBS classes. These figures compare extremely favorably with past history in which the average waits exceed five weeks.

APPENDIX A
OPTIMIZATION TECHNIQUES FOR SOLVING THE PROBLEM

There were 3 optimization techniques considered: *mixed integer programming* (*MIP*), *Lagrangean relaxation* and *Benders decomposition.*

1. MIXED INTEGER PROGRAMMING

The mixed integer programming model of the Marine Corps problem is presented in the following formulation. This model exploits the integrality enforcing capability by using it as a selection mechanism to develop the course schedule for MOS schools with discretionary start dates. The MIP formulation discretizes time which is measured in weekly units. For notational convenience, the following sets are defined :

(a) M1 MOS's with non-USMC controlled MOS school

(b) M2 MOS's with USMC controlled MOS school

(c) M3 MOS's with USMC controlled MOS school excluding Infantry

(d) M4A MOS's belonging to set M1 which have warrant officers and unrestricted officers in the same MOS classes

(e) M4B MOS's belonging to set M2 which have warrant officers and unrestricted officers in the same MOS classes

(f) TU TBS classes for unrestricted officers

(g) XP Possible assignments from OCS to TBS classes[6]

(h) YP1 Possible assignments from TBS to non-USMC controlled MOS classes

(i) YP2 Possible assignments from TBS to USMC controlled MOS classes

[6]That is waiting time has to be non-negative

Formulation of Mixed Integer Programming Model :

Indices :

h	OCS classes	$h = 1,2,3$
i	TBS classes	$i = 1,2...9$
k	Types of Ground MOS's	$k = 1,2...21$
j	Potential start weeks for MOS classes	$j = 1,2...52$
n	MOS class numbers (applies only to USMC controlled MOS schools)	$n = 1,2...N_k$
t	alias for j	

Parameters :

(a) *Quotas*

Q_k	Output for MOS k
P_k	Previous FY's TBS graduates assigned to classes held in the current FY by MOS course k
WO_k	Number of warrant officers assigned to MOS k
F_k	Number of Air officers rechannelled to MOS k
V_k	Number of officers on voluntary lateral move into MOS k
D_k	Number of officers on directed lateral move into MOS k
GO_h	Number of Ground OCC officers graduating from OCS class h
AO_h	Number of Air OCC officers graduating from OCS class h
AT_i	Number of non-OCC Air officers entering TBS class i

(b) *Training school restrictions*

LC_k	Lower class size limit in USMC controlled MOS course k
UC_k	Upper class size limit in USMC controlled MOS course k
DM_k	Class duration for USMC controlled MOS course k
ME_k	End date for last class in the previous FY conducted by USMC controlled MOS course k
NST_{kn}	Number of seats available in class n of non USMC controlled MOS course k
N_k	Number of classes open to USMC officers in non-USMC controlled MOS course k
MIN_k	Minimum assignment from each TBS class to MOS k
LV_k	Leave period after TBS training for MOS k
OJT_k	OJT period after TBS training for MOS k

(c) *Waiting times*

W_{hi}	Wait for officer assigned from OCS class h to TBS class i
$W1_{ikn}$	Wait for officer assigned from TBS class i to class n of non-USMC controlled MOS course k
$W2_{ikj}$	Wait for officer assigned from TBS class i to class held in week j by USMC controlled MOS course, provided week j is selected to hold a class

(d) *Other*

W Class number for warrant officer TBS class

Decision Variables :

(a) *Assignments into TBS*

- GT_i Number of non-OCC Ground Officers assigned to TBS class i
- AX_{hi} Air OCC officer assignments from OCS class h to TBS class i
- GX_{hi} Ground OCC officer assignments from OCS class h to TBS class i

(b) *Assignments from TBS to MOS courses*

- $Y1_{ikn}$ Assignment of Ground officers from TBS class i to class n of non-USMC controlled MOS course k
- $Y2_{ikj}$ Assignment of Ground officers from TBS class i to class held in week j by USMC controlled MOS course k

(c) *Assignments from non-TBS sources to MOS* courses[7]

- $P1_{kn}$ Assignment of previous FY's TBS graduates into class n of non-USMC controlled MOS course k
- $P2_{kj}$ Assignment of previous FY's TBS graduates to class held in wk j by USMC controlled MOS course k
- $V1_{kn}$ Assignment of voluntary lateral move officers to class n of non-USMC controlled MOS course k
- $V2_{kj}$ Assignment of voluntary lateral move officers to class held in wk j by USMC controlled MOS course k
- $D1_{kn}$ Assignment of directed lateral move officers to class n of non-USMC controlled MOS course k
- $D2_{kj}$ Assignment of directed lateral move officers to class held in wk j by USMC controlled MOS course k
- $F1_{kn}$ Assignment of rechannelled Air officers to class n of non-USMC controlled MOS course k
- $F2_{kj}$ Assignment of rechannelled Air officers to class held in wk j by USMC controlled MOS course k

(d) *Class sizes*

- TBS_i Size of TBS class i
- $MS1_{kn}$ Number of USMC candidates in class n of non-USMC controlled MOS course k
- $MS2_{kj}$ Size of class held in wk j by USMC controlled MOS course k

(e) *Binary variables*

- B_{kj} Binary variable used to indicate class held in wk j for MOS k

[7] This includes assignments from the previous FY's TBS classes

Mathematical Formulation (P1) :

$$\text{Min} \quad \sum_{h}\sum_{i} W_{hi} * (AX_{hi} + GX_{hi}) + \sum_{\substack{k \\ k\in M1 \\ (i,n)\in YP1}}\sum_{i}\sum_{n} W1_{ikn} * Y1_{ikn}$$
$$\qquad (h,i)\in XP$$

$$+ \sum_{\substack{k \\ k\in M2 \\ (i,j)\in YP2}}\sum_{i}\sum_{j} W2_{ikj} * Y2_{ikj} \qquad (0)$$

s.t.

$$\sum_{\substack{i \\ (h,i)\in XP}} AX_{hi} = AO_h, \quad h = 1,2,3 \qquad (1)$$

$$\sum_{\substack{i \\ (h,i)\in XP}} GX_{hi} = GO_h, \quad h = 1,2,3 \qquad (1a)$$

$$\sum_{\substack{h \\ (h,i)\in XP}} (AX_{hi} + GX_{hi}) + GT_i + AT_i = TBS_i, \quad i \in TU \qquad (2)$$

$$\sum_{\substack{k \\ k\in M1 \\ (i,n)\in YP1}}\sum_{n} Y1_{ikn} + \sum_{\substack{k \\ k\in M2 \\ (w,j)\in YP2}}\sum_{j} Y2_{ikj} + \sum_{\substack{h \\ (h,i)\in XP}} AX_{hi} + AT_i = TBS_i, \quad i \in TU \qquad (3)$$

$$\sum_{\substack{n \\ (w,n)\in YP1}} Y1_{wkn} = WO_k, \quad k \in M4A \qquad (4)$$

$$\sum_{\substack{j \\ (w,j)\in YP2}} Y2_{wkj} = WO_k, \quad k \in M4B \qquad (4a)$$

$$\sum_{\substack{i \\ (i,n)\in YP1}} Y1_{ikn} + P1_{kn} + V1_{kn} + D1_{kn} + F1_{kn} = MS1_{kn}, \quad k \in M1 \text{ and } n = 1,2 \ldots N_k \qquad (5)$$

$$\sum_{\substack{i \\ (i,j)\in YP2}} Y2_{ikj} + P2_{kj} + V2_{kj} + D2_{kj} + F2_{kj} = MS2_{kj}, \quad k \in M2 \text{ and } j = 1,2..52 \qquad (5a)$$

$$\sum_{n} MS1_{kn} = Q_k + P_k, \quad k \in M1 \qquad (6)$$

$$\sum_{j} MS2_{kj} = Q_k + P_k, \quad k \in M2 \qquad (6a)$$

$$\sum_{n} P1_{kn} = P_k, \quad k \in M1 \qquad (7)$$

$$\sum_{j} P2_{kj} = P_k, \quad k \in M2 \qquad (7a)$$

$$\sum_{n} V1_{kn} = V_k, \quad k \in M1 \qquad (8)$$

$$\sum_{j} V2_{kj} = V_k, \quad k \in M2 \qquad (8a)$$

$$\sum_{n} D1_{kn} = D_k, \quad k \in M1 \qquad (9)$$

$$\sum_{j} D2_{kj} = D_k, \quad k \in M2 \qquad (9a)$$

$$\sum_{n} F1_{kn} = F_k, \quad k \in M1 \qquad (10)$$

$$\sum_j F2_{kj} \quad = \quad F_k, \qquad k \in M2 \qquad (10a)$$

$$\sum_{j=t}^{t+DM_k} B_{kj} \quad \leq \quad 1, \qquad k \in M3 \text{ and } t=1,..52 \qquad (11)$$

$$\sum_{\substack{n \\ (i,n) \in YP1}} Y1_{ikn} \quad \geq \quad MIN_k, \qquad i \in TU \text{ and } k \in M3 \qquad (12)$$

$$\sum_{\substack{j \\ (i,j) \in YP2}} Y2_{ikj} \quad \geq \quad MIN_k, \qquad i \in TU \text{ and } k \in M4 \qquad (12a)$$

$$LC_k * B_{kj} \quad \leq \quad MS2_{kj}, \qquad k \in M2 \text{ and } j=1,..52 \qquad (13)$$

$$UC_k * B_{kj} \quad \geq \quad MS2_{kj}, \qquad k \in M2 \text{ and } j=1,..52 \qquad (14)$$

Variable bounds:

$B_{kj} \in (0,1), \quad k \in M2$ and $j=1,...52$

$B_{kj} = 0$ if $j \leq ME_k, \quad k \in M3$ and $j=1,...52$

$150 \leq TBS_i \leq 250, \quad i \in TU$

$0 \leq MS1_{kn} \leq NST_{kn}, \quad k \in M1$ and $n=1,..N_k$

The terms of the objective function (0) represent waiting time (in man weeks) incurred by the different groups of class assignments : OCS to TBS, TBS to MOS for officers attending non-USMC controlled MOS courses, and TBS to MOS for officers attending USMC controlled MOS courses. Class start dates for USMC controlled MOS courses are decision variables. Every week of the current FY represents a potential class start date for each of these MOS courses. The model is formulated so that assignments can be made to all classes that may potentially be scheduled. For non-USMC controlled MOS courses, their class start dates are fixed and assignments can be made only to classes that are already scheduled.

Constraints (1) to (10a) are flow balance equations. Constraints (1) and (1a) preserve the OCS class throughput for Air and Ground officers respectively. Constraints (2) sum the number of officers entering each unrestricted officer TBS class i and equates it to the TBS class size variable TBS_i. Constraints (3) sum the number of officers leaving the unrestricted officer TBS class i and equates it to TBS_i so as to preserve the flow of officers through the TBS class. Constraints (4) and (4a) ensures that all warrant officers attending mixed MOS classes have been assigned for both non-USMC and USMC controlled MOS's respectively. Constraints (5) and (5a) preserve the MOS class throughput for non-USMC and USMC controlled MOS's respectively.

Constraints (6) and (6a) force the MOS output requirements to be fulfilled for non-USMC and USMC controlled MOS's respectively. Constraints (7) to (10a) ensure that all officers from the following respective groups are assigned to MOS classes : Previous FY's TBS output, voluntary lateral movements, directed lateral movements and grounded Air officers.

Constraints (11) form the restriction of not allowing overlapping MOS classes for all USMC controlled MOS's except the Infantry MOS. If week j is selected for a class, then no other class will be scheduled until this class is completed. Constraints (12) and (12a) forces each TBS class to assign the the prescribed minimum number of officers to non-USMC controlled and USMC controlled MOS's respectively. Classes for the latter MOS's have lower and upper bounds if open as expressed in constraints (13) and (14).

All variables are non-negative, including some which are binary. The binaries perform a go no-go function to indicate if a particular week has been selected to schedule a class for those schools with variable class start dates. If MOS k disallows the scheduling of overlapping classes, then B_{kj} is initialized to zero whenever the last MOS class from the previous year overlaps week j.

There is a problem with implementing the above formulation on a computer. In general, large integer programs are difficult to solve. As the number of integer variables grow, the solution time of the formulation may increase dramatically. Formulation (P1) has a total of 780 integer (binary) variables because there are 52 binary variables for each of the 15 USMC controlled MOS's. Such a large problem is not routinely solved with MIP. Through the courtesy of Dr Alexander Meeraus, one of the co-developers of the GAMS Modeling system, an unsuccessful attempt was made to solve a reduced formulation on the World Bank's IBM 3090 mainframe computer. The reduced formulation considered the problem with eight arbitrarily selected USMC MOS courses.

2. LAGRANGEAN RELAXATION

As an alternative, one can consider using *Lagrangean relaxation*, a popular technique used successfully on many occasions to solve difficult integer problems. The idea behind Lagrangean relaxation stems from the observation that many difficult integer problems are relatively easy to solve when stripped of those constraints that complicate them. The method has been widely covered in literature, including some well-written articles by Fisher [Ref. 7,8]. The following example was quoted from one of these articles.

Consider the following integer problem:

$$(P2) \quad V = \operatorname{Min} \; cx$$
$$\text{s.t.} \quad \mathbf{Hx} \le \mathbf{b}$$
$$\mathbf{Gx} \le \mathbf{e}$$
$$\mathbf{x} \ge \mathbf{0} \text{ and integral.}$$

where $\mathbf{x}$ is n x 1, $\mathbf{b}$ is m x 1, $\mathbf{e}$ is k x 1, and all other matrices have conformable dimensions.

The constraints of (P2) are partitioned into two sets $\mathbf{Hx} \le \mathbf{b}$ and $\mathbf{Gx} \le \mathbf{e}$ to make it easy to solve the Lagrangean problem :

$$(P3) \quad V_D(\mathbf{u}) = \operatorname{Min} \; cx + \mathbf{u}(\mathbf{e} - \mathbf{Gx})$$
$$\text{s.t.} \quad \mathbf{Hx} \le \mathbf{b}$$
$$\mathbf{x} \ge \mathbf{0} \text{ and integral.}$$

For $\mathbf{u} \ge \mathbf{0}$, the optimal value for problem (P3) forms a lower bound for V.

The best lower bound is obtained by solving the dual problem

$$(D) \quad \max_{u \ge 0} \; V_D(\mathbf{u})$$

A relaxation for formulation (P1) can be obtained by dualizing constraints (12), (12a), (13) and (14). As shall be shown, the resulting formulation after removal of these constraints is a network problem that can be efficiently solved using a readily available network solver.

The constraint matrix for constraints (1) to (10a) is unimodular implying they form part of a network. Venoitt et al [Ref. 9] and Rosenthal [Ref. 10] show a method by which constraints (11) can be reduced to possess the same constraint matrix property. The method uses elementary row operations as the case for MOS *k* demonstrates:

The original equations are (*after adding in slack variables*)

$$
\begin{array}{lllllll}
B_{1k} & + \dots & B_{dk} & + S_{1k} & & & = 1 \\
& B_{2k} & + \dots & & + S_{2k} & & = 1 \\
& & \dots\dots\dots\dots & & & & \\
& & B_{52-d,k} & + \dots & & + S_{52-d,k} & = 1
\end{array}
$$

where *d* is the duration of the MOS class.

Except for the first row, subtract each row from the preceeding row :

$$
\begin{array}{llllll}
B_{1k} & + \ldots & B_{dk} & + S_{1k} & & = 1 \\
& -B_{2k} + B_{d+1,k} & & - S_{1k} & + S_{2k} & = 0 \\
& \ldots\ldots\ldots\ldots & & & & \\
& -B_{51-d,k} & + B_{52,k} & - S_{51-d,k} & + S_{52-d,k} & = 0
\end{array}
$$

By following the above computation, it will be seen that each column in the constraint matrix has at most two non-zero elements and if two, they are +1 and -1. The matrix is thus unimodular.

To formulate the Lagrangean problem for (P1), for simplification, the matrix notation $\mathbf{Ay} \leq \mathbf{b}$ is being used instead of constraints (12) to (14). The Lagrangean problem is :

$$
\text{Min} \quad \sum W_{hi} * (AX_{hi} + GX_{hi}) + \sum\sum W1_{ikn} * Y1_{ikn} + \sum\sum W2_{ikj} * Y2_{ikj} + \mathbf{u}\,(\,\mathbf{b} - \mathbf{yA}\,)
$$

subject to constraints (1) to (11) and the same variable bounds as (P1).

The optimal solution for any choice of $\mathbf{u} \geq \mathbf{0}$ yields a lower bound to (P1).The best lower bound is obtained by solving the dual problem

$$
(\text{P4}) \qquad \underset{u \geq 0}{\text{Max}}\; V_D\,(\mathbf{u})
$$

It is clear that the solution algorithm will have to determine **u** which optimizes or closely optimizes (P4). The vector **u** has an element for each "difficult" constraint. It can be checked that this vector will have approximately 1916 elements based on 168 elements each for constraints (12) and (12a), and 780 elements each for constraints (13) and (14). Such a large **u** vector will make it extremely difficult to use this method.

In general, the $V_D(\mathbf{u})$ function is convex and differentiable except at points where the Lagrangean problem has multiple optima. These properties make it attractive to utilize a gradient-based hill climbing method. With multiple optimal solutions, the function is non-differentiable and requires an adaptation rule for tie-breaking.

3. BENDERS DECOMPOSITION

This technique was first proposed by the person after whom it was named in 1962 [Ref. 11]. The decomposition principle provides a systematic procedure for successively solving a *sub-problem* and a *master problem* until the optimum is achieved and verified. For mixed integer problems, the sub-problem is formed from the original

problem by fixing the values for all integer variables, and the master problem by relaxation of the original problem through removal of "difficult" constraints. During each iteration of the decomposition algorithm, the fixed variables in the sub-problem are adjusted by the master problem.

The method is explained using the following notation provided by Van Roy [Ref. 12]. The primal problem is:

$$(\mathrm{P}) \qquad \min_{x \in S} \; cx$$

$$\text{s.t.} \quad \mathbf{Ax} \geq \mathbf{b}$$

where $\mathbf{x}$ is n x 1, $\mathbf{A}$ is m x n, and $\mathbf{c}$ and $\mathbf{b}$ have conformable dimensions; Set S is a subset of R^n and restricts some elements in $\mathbf{x}$ to be integral,

i.e. $S \equiv (\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T, \mathbf{x}_1 \geq \mathbf{0}, \mathbf{x}_2 \in Z)$ where Z is a subset of integers.

By partitioning matrix $\mathbf{A}$ such that $m = m_1 + m_2$ and $n = n_1 + n_2$, the original problem can be notationally expanded to

$$(\mathrm{P}) \qquad \min_{\mathbf{x} \in S} \; \mathbf{c}^1\mathbf{x}_1 + \mathbf{c}^2\mathbf{x}_2$$

$$\text{s.t.} \quad \begin{aligned} \mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2 &\geq \mathbf{b}_1 \\ \mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 &\geq \mathbf{b}_2 \end{aligned}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}^1 & \mathbf{A}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$

and $\mathbf{A}_2\mathbf{x} \geq \mathbf{b}_2$ are the complicating constraints of (P). By definition, the relaxation of (P) through removal of constraints $\mathbf{A}_2\mathbf{x} \geq \mathbf{b}_2$ and the restriction of (P) by fixing values for variables $\mathbf{x}_2$ are problems that will be relatively easy to solve.

(P) can be rewritten as:

$$(\mathrm{P}) \qquad \min_{\mathbf{x}_2 \in Z} \begin{bmatrix} \displaystyle\min_{\mathbf{x}_1 \geq 0} \; \mathbf{c}^1\mathbf{x}_1 + \mathbf{c}^2\mathbf{x}_2 \\ \\ \text{s.t.} \quad \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 \geq \mathbf{b} \end{bmatrix}$$

or

$$\min_{x_2 \in Z} \left[\begin{array}{ll} \text{(PD)} & \\ \underset{u \geq 0}{\text{Max}} & u(b - A^2 x_2) + c^2 x_2 \\ & \\ \text{s.t.} & uA^1 \leq c_1 \end{array} \right]$$

or

(MP) $\quad \underset{x_2 \in Z, x_o}{\text{Min}}\ x_o$

$\quad$ s.t. $u^j b + (c^2 - u^j A^2)\, x_2 \leq x_o, \quad j = 1,2....T$

where

(SP) $\quad \underset{x_1 \geq 0}{\text{Min}}\ c^1 x_1 + c^2 x_2$

$\quad$ s.t. $A^1 x_1 \geq b - A^2 x_2$

and u^j, j=1,2....T are extreme points in the feasible region of the dual(PD) to the primal sub-problem. x_2 has been excluded in the feasible region to (PD) for simplicity since the latter does not depend on x_2. The master program(MP) is a MIP problem with only 1 continous variable x_o. The constraints in (MP) are called *Benders cuts.* There are a total of T constraints in (MP),i.e.one for each extreme point of the feasible region in (PD). Evidently, solving (MP) is equivalent to solving the original formulation (P) but it requires all extreme points explicitly.

Briefly, the steps of Benders' algorithm are:

(a) Set B_u = M where M is an arbitrary large number. Select some u^j which is feasible for (PD).

(b) Solve (MP) using u^j from step(a).
Let x_2, x_o be the solutions.

(c) Generate the most violated constraint for (MP) by solving (PD) using the solutions from step(b).
Let u^{j+1}, u_o(*objective value*) be the solutions.

(d) Let B_u = Min (B_u, u_o). If $x_o \geq B_u - \epsilon$ where ϵ is the convergence criteria, then stop.
Otherwise, add the constraint $u^{j+1} b + (c^2 - u^{j+1} A^2)\, x_2 \leq x_o$ to (MP).
Return to step(b).

Some encouraging results have been reported with this technique. The successes come about when using variations of canonical Benders' decomposition. Further

research is necessary before it is possible to assess whether the method can be applied to solve the Marine Corps problem successfully.

APPENDIX B
SAMPLE INPUT FILE FOR SCHED FORTRAN PROGRAM

(File format is given in Appendix H)

```
***   SECTION I -- OCS ENTRIES    ***
START    OCC(AIR)
DATE     OFFRS
  5         8         OCS1
 21        20         OCS2
 46        28         OCS3
***   SECTION II -- TBS ENTRIES    ***
END       START    NON-OCC(AIR)
DATE      DATE     OFFRS
 39        15        54         TBS1
 44        20        60         TBS2
 52        26        29         TBS3
 57        31        37         TBS4
 62        36        89         TBS5
 74        50        46         TBS6
 79        56         0         TBS7
 86        62        25         TBS8
* WARRANT OFFICER INPUTS *
CLASS      END
NUMBER     DATE
  7         65
 ***    SECTION III -- MOS ENTRIES    ***
DUR  MIN   Q   NTBS   WO   OJT   LV PREFY  NM  PREIND LSIZE   MOS
 19   6   125    0     0    0     2   39    8    0     99     ARTY
 15   1    23    0     0    0     2   39    5    0     99     TANK
 18   1    19    7     0    0     2   39   10    0     99     ADA
  9   1    18    3     2   12     0   39    7    0     99     AMO
  7   0     2    3     0    0     2   39    4    0     99     PAO
  8   0     1    9     0    0     2   39    8    0     99     MPO
  9  12   254    5     0    0     0   39   99    0     30     INFAN
 14   2    85   15     0    0     2   39   99    0     30     COMM
 12   2    80    5     5    0     2   39   99    0     15     SUP
 10   2    54    9     0    0     2   39   99    0     13     ASUP
  7   2    53    2     7    0     2   39   99    0     20     MT
 12   2    50    5     0    0     2   39   99    0     25     LOGS
 12   2    43    0     0    0     2   39   99    0      7     ENG
 18   2    33   10     1    0     2   39   99    0     15     FIN
 13   1    26    8     0    0     2   39   99    0      6     ADC
 15   1    26    5     8   12     0   39   99    0     10     AVNSP
 13   1    19    3     2    0     2   39   99    0     10     ADP
 18   1    18    7     0    0     2   39   99    0      7     ATC
  8   1    14    3     0    0     2   39   99    0      6     AMPH
  6   0     5   40     0    0     2   39   99    0     20     ADJ
  9   0     2   55    12    0     2   45   99    0     15     INT
* MOS VARIABLE DEFINITIONS *
DUR         =   DURATION OF MOS CLASS
MIN         =   MINIMUM ASSIGNMENT FROM TBS CLASS TO MOS (5% OF Q)
Q           =   MOS OUTPUT FROM TBS CLASSES
NTBS        =   MOS OUTPUT FROM NON-TBS SOURCES
WO          =   NUMBER OF WO'S IN MOS WITH "MIXED" MOS CLASSES
OJT         =   ON-THE-JOB TRAINING PERIOD AFTER TBS (IN WEEKS)
LV          =   ENFORCED LEAVE PERIOD AFTER TBS (IN WEEKS)
PREFY       =   ENDING DATE OF LAST MOS CLASS SCHEDULED IN PREVIOUS FY
NM          =   NUMBER OF CLASSES (ENTER 99 FOR USMC MOS)
PREIND      =   ENTER 1 IF MOS HAS PRESELECTED DATES; ELSE ENTER 0
LSIZE       =   MINIMUM CLASS SIZE (ENTER 99 FOR NON-USMC MOS)
*  START DATES OF NON-USMC COURSES  *
46   49   54   59   65   73   85   92                  ARTY
48   62   77   83   92                                TANK
42   50   56   61   65   71   75   80   84   88       ADA
```

```
55  58  66  74  78  81  98                              AMO
46  56  65  75                                          PAO
49  53  59  63  66  72  76  81                          MPO
 *  SEAT ALLOCATIONS FOR NON-USMC COURSES  *
11  12  12  16  25  13  12  24                          ARTY
 5   5   4   3   6                                      TANK
 4   2   4   4   3   2   2   2   2   1                  ADA
 3   3   3   3   3   3   5                              AMO
 1   2   1   1                                          PAO
 1   2   1   1   1   2   1   1                          MPO
  ***     SECTION IV -- PRESELECTED COURSE ENTRIES   ***
 NUMBER           START DATES
 OF CLASSES       OF MOS COURSE WITH PRESELECTED SCHEDULE
    8             45   50   67   70   75   80   85   88   MOS 1
    3             45   67   76                             MOS 2
    7             43   50   58   73   75   81   85        MOS 3
    4             45   67   76   89                        MOS 4
    3             41   67   88                             MOS 5
```

APPENDIX C
SAMPLE FOR FILE SCHED.GMS

```
TABLE STMOS(K,J) STARTING WEEK OF MOS CLASS
         C1     C2     C3     C4     C5     C6     C7     C8     C9     C10
  ARTY   46     49     54     59     65     73     85     92
  TANK   48     62     77     83     92
  ADA    42     50     56     61     65     71     75     80     84     88
  AMO    55     58     66     74     78     81     98
  PAO    46     56     65     75
  MPO    49     53     59     63     66     72     76     81
  INFAN  39     44     52     57     62     74     79     86
  COMM   54     76     90
  SUP    41     54     66     81     93
  AIRSUP 41     54     64     81     93
  MT     41     54     81     93
  LOGS   54     81    106
  ENG    46     59     71     83     98
  FIN    59     88
  ADC    50     63     76     89
  AVNSP  56     74     91    108
  ADP    46     76     98
  ATC    54     81    106
  AMPH   59     88
  ADJ    46     76     98
  INT    46     59     76     88
PARAMETER NUM(K) NUMBER OF CLASSES PER YEAR FOR MOS K
  ARTY    8
  TANK    5
  ADA    10
  AMO     7
  PAO     4
  MPO     8
  INFAN   8
  COMM    3
  SUP     4
  AIRSUP  4
  MT      3
  LOGS    2
  ENG     4
  FIN     2
  ADC     4
  AVNSP   3
  ADP     2
  ATC     2
  AMPH    2
  ADJ     2
  INT     4
PARAMETER MAX(K)   NUMBER OF CLASSES SCHEDULED FOR MOS K
  ARTY    8
  TANK    5
  ADA    10
  AMO     7
  PAO     4
  MPO     8
  INFAN   8
  COMM    3
  SUP     5
  AIRSUP  5
  MT      4
  LOGS    3
  ENG     5
  FIN     2
```

```
     ADC      4
     AVNSP    4
     ADP      3
     ATC      3
     AMPH     2
     ADJ      3
     INT      4
PARAMETER MWAIT(K)     MAXIMUM WAIT BETWEEN TBS AND MOS CLASSES
     ARTY     9
     TANK    13
     ADA     11
     AMO     12
     PAO     11
     MPO      8
     INFAN    9
     COMM    17
     SUP     14
     AIRSUP  14
     MT      22
     LOGS    22
     ENG     10
     FIN     24
     ADC     12
     AVNSP   14
     ADP     22
     ATC     22
     AMPH    24
     ADJ     22
     INT     12
```

TABLE NEAREST(K,I) WAIT FROM TBS TO NEAREST FOLLOW-ON MOS CLASS

	TBS1	TBS2	TBS3	TBS4	TBS5	TBS6	TBS7	TBS8	TBS9
ARTY	5	0	0	0	1	9	6	4	4
TANK	7	2	8	3	13	1	10	2	4
ADA	1	4	2	2	1	4	4	3	0
AMO	4	2	2	5	0	12	1	7	0
PAO	5	0	2	6	1	0	8	0	0
MPO	8	3	5	0	2	0	5	0	0
INFAN	0	0	0	0	0	0	9	0	0
COMM	13	8	0	17	12	0	9	9	2
SUP	0	8	0	7	2	5	14	0	5
AIRSUP	0	8	0	5	0	5	14	0	5
MT	0	8	0	22	17	5	14	0	5
LOGS	13	8	0	22	17	5	14	0	18
ENG	5	0	5	0	7	7	4	2	10
FIN	18	13	5	0	24	12	21	7	0
ADC	9	4	9	4	12	0	9	8	1
AVNSP	5	0	10	5	0	5	14	0	10
ADP	5	0	22	17	12	0	9	17	10
ATC	13	8	0	22	17	5	14	0	18
AMPH	18	13	5	0	24	12	21	7	0
ADJ	5	0	22	17	12	0	9	17	10
INT	5	0	5	0	12	0	9	7	0

APPENDIX D
SOURCE CODE FOR ASSIGN GAMS PROGRAM

```
$TITLE OFFICER ASSIGNMENT MODEL
$OFFUPPER
*  Developed by : K. S. CHNG, Naval Postgraduate School, Sep 1987.
******************************************************************
*  NOTE   :A. URO DENOTE UNRESTRICTED OFFICERS                  *
*          B. TBS7 IS RESERVED CLASS FOR WO CANDIDATES          *
******************************************************************

SETS

  H  OCS COURSES /OCS1 * OCS3/
  I  TBS COURSES /TBS1 * TBS9/
  K  TYPES OF MOS
    / ARTY        Artillery
      TANK        Tank
      ADA         Anti-Aircraft
      AMO         Air Maintenance
      PAO         Public Affairs
      MPO         Military Police
      INFAN       Infantry
      COMM        Communications
      SUP         Supply(Ground)
      AIRSUP      Air Support
      MT          Motor Transport
      LOGS        Logistics
      ENG         Combat Engineer
      FIN         Finance
      ADC         Air Defence Control
      AVNSP       Aviation Support
      ADP         Data Processing
      ATC         Air Traffic Control
      AMPH        Amtrack
      ADJ         Adjutant
      INT         Intelligence/

  J  MOS COURSES / C1 * C10/

PARAMETER   QUOTA(k)  Yearly output quota for MOS k
   /  ARTY        125
      TANK         23
      ADA          26
      AMO          23
      PAO           5
      MPO          10
      INFAN       259
      COMM        100
      SUP          90
      AIRSUP       63
      MT           62
      LOGS         55
      ENG          43
      FIN          44
      ADC          34
      AVNSP        39
      ADP          24
      ATC          22
      AMPH         14
      ADJ          45
      INT          69/

PARAMETER  LEFTOVER(k)  TBS output brought fwd from previous yr
   /  ARTY          0
```

```
          TANK          0
          ADA           0
          AMO           0
          PAO           0
          MPO           0
          INFAN         0
          COMM          0
          SUP           0
          AIRSUP        0
          MT            0
          LOGS          0
          ENG           0
          FIN           0
          ADC           0
          AVNSP         0
          ADP           0
          ATC           0
          AMPH          0
          ADJ           0
          INT           0/
PARAMETER  WOMOS(k)  WOs to be assigned to mixed MOS classes
     / AMO          2
          SUP           5
          MT            7
          FIN           1
          AVNSP         8
          ADP           2
          INT          12 /
SCALAR  W          Warrant officer TBS class number / 7/
          OCSATT   OCS attrition rate            / .45/
          MAXLEFT  Maximum waiting time for Leftovers/30/
          TL       Lower capacity of TBS class   /150/
          TU       Upper capacity of TBS class   /250/
SETS MOS1(k)  MOS with no leave after TBS / AMO,INFAN,AVNSP/
      MOS2(k)  MOS with OJT after TBS /AMO,AVNSP/
      MOS3(k)  Non-USMC controlled MOS schools
             / ARTY,TANK,ADA,AMO,PAO,MPO/
      MOSWO(k) MOS with mixed warrant offr and unrestricted offr classes
             / AMO,SUP,MT,FIN,AVNSP,ADP,INT/
SETS MOS4  MOS with fixed leave period after TBS;
     MOS4(k)       = YES ;
     MOS4(MOS1)    = NO  ;
SET MOS5  USMC controlled MOS schools;
* Difference of sets k and MOS3
     MOS5(k)       = YES ;
     MOS5(MOS3)    = NO  ;
SET TBSURO(i) TBS classes for unrestricted officers;
     TBSURO(i)  = YES $ (ORD(i) NE W)

PARAMETER LEAVE(k)  Fixed leave after TBS ;
     LEAVE(k) $ (MOS4(k))  = 2 ;
PARAMETER OJT(k)  On-the-Job training period after TBS ;
     OJT(k) $ (MOS2(k)) = 12 ;

PARAMETER FLT(k) FLT attrite & NPQ & FFPB assigned to MOS k
    /  ARTY         0
          TANK          0
          ADA           5
          AMO           3
          PAO           1
          MPO           6
          INFAN         5
          COMM          5
          SUP           5
          AIRSUP        5
```

```
            MT               0
            LOGS             5
            ENG              0
            FIN              5
            ADC              5
            AVNSP            5
            ADP              0
            ATC              5
            AMPH             0
            ADJ             25
            INT             15/
PARAMETER INTTOT(k) Intended MOS offrs assigned to MOS k
      /    ARTY             0
            TANK             0
            ADA              0
            AMO              0
            PAO              0
            MPO              0
            INFAN            0
            COMM             0
            SUP              0
            AIRSUP           0
            MT               0
            LOGS             0
            ENG              0
            FIN              0
            ADC              0
            AVNSP            0
            ADP              0
            ATC              0
            AMPH             0
            ADJ              0
            INT              0/

PARAMETER VOLLAT(k) Expected number of Vol LATMOVs into MOS k
      /    ARTY             0
            TANK             0
            ADA              2
            AMO              0
            PAO              2
            MPO              3
            INFAN            0
            COMM            10
            SUP              0
            AIRSUP           4
            MT               2
            LOGS             0
            ENG              0
            FIN              5
            ADC              3
            AVNSP            0
            ADP              3
            ATC              2
            AMPH             0
            ADJ              5
            INT             10/

PARAMETER INDLAT(k)  Directed LATMOVS assigned to MOS k
      /    ARTY             0
            TANK             0
            ADA              0
            AMO              0
            PAO              0
            MPO              0
            INFAN            0
            COMM             0
            SUP              0
            AIRSUP           0
```

```
        MT              0
        LOGS            0
        ENG             0
        FIN             0
        ADC             0
        AVNSP           0
        ADP             0
        ATC             0
        AMPH            0
        ADJ            10
        INT            30/
PARAMETER  GOCC(h)  Grd OCC officers entering OCS class h
     / OCS1       92
        OCS2      80
        OCS3      72/

PARAMETER  AOCC(h)  Air OCC Officers entering OCS class h
     / OCS1        8
        OCS2      20
        OCS3      28/
*  Sum of AOCC h and GOCC h must be between 100 and 150 for all h
PARAMETER GOCCPASS(h)  Grd OCC output from OCS class h;
  GOCCPASS(h)  = CEIL(GOCC(h) * (1- OCSATT))  ;
PARAMETER AOCCPASS(h)  Air OCC output from OCS class h;
  AOCCPASS(h)  = CEIL(AOCC(h) * (1- OCSATT))  ;
PARAMETER   ATBS(i)  Air Z Officers entering TBS class i
     / TBS1       54
        TBS2      60
        TBS3      29
        TBS4      37
        TBS5      89
        TBS6      46
        TBS8       0
        TBS9      25/
TABLE  NSEAT(k,j)  Reserved seats at non-USMC MOS schools
                      C1   C2   C3   C4   C5   C6   C7   C8   C9   C10
        ARTY          11   12   12   16   25   13   12   24
        TANK           5    5    4    3    6
        ADA            4    2    4    4    3    2    2    2    2     1
        AMO            3    3    3    3    3    3    5
        PAO            1    2    1    1
        MPO            1    2    1    1    1    2    1    1
PARAMETER NMIN(k)  Lower capacity for MOS class
    /  INFAN          30
        COMM          30
        SUP           15
        AIRSUP        13
        MT            20
        LOGS          25
        ENG            7
        FIN           15
        ADC            6
        AVNSP         10
        ADP           10
        ATC            7
        AMPH           6
        ADJ           20
        INT           15  /
PARAMETER NMAX(k)  Upper capacity for MOS class
    /  INFAN          50
        COMM          49
        SUP           31
        AIRSUP        17
        MT            28
        LOGS          40
```

```
               ENG          20
               FIN          31
               ADC          12
               AVNSP        15
               ADP          20
               ATC          16
               AMPH         20
               ADJ          35
               INT          33 /


PARAMETER STOCS(h)  Starting week of OCS class h
     / OCS1        5
       OCS2       21
       OCS3       46/

PARAMETER  STTBS(i)  Starting week of TBS class i
     / TBS1        15
       TBS2        20
       TBS3        26
       TBS4        31
       TBS5        36
       TBS6        50
       TBS7        52
       TBS8        56
       TBS9        62/

PARAMETER ENDTBS(i)  Ending week of TBS class
     / TBS1        39
       TBS2        44
       TBS3        52
       TBS4        57
       TBS5        62
       TBS6        74
       TBS7        65
       TBS8        79
       TBS9        86/


 TABLE STMOS(K,J) Starting week of MOS class
          C1    C2    C3    C4    C5    C6    C7    C8    C9    C10
   ARTY   46    49    54    59    65    73    85    92
   TANK   48    62    77    83    92
   ADA    42    50    56    61    65    71    75    80    84    88
   AMO    55    58    66    74    78    81    98
   PAO    46    56    65    75
   MPO    49    53    59    63    66    72    76    81
   INFAN  39    44    52    57    62    74    79    86
   COMM   54    76    90
   SUP    41    54    66    81    93
   AIRSUP 41    54    64    81    93
   MT     41    54    81    93
   LOGS   54    81    106
   ENG    46    59    71    83    98
   FIN    59    88
   ADC    50    63    76    89
   AVNSP  56    74    91    108
   ADP    46    76    98
   ATC    54    81    106
   AMPH   59    88
   ADJ    46    76    98
   INT    46    59    76    88
PARAMETER NUM(k) Number of classes per year for MOS k
/  ARTY    8
   TANK    5
   ADA    10
   AMO     7
```

```
    PAO      4
    MPO      8
    INFAN    8
    COMM     3
    SUP      4
    AIRSUP   4
    MT       3
    LOGS     2
    ENG      4
    FIN      2
    ADC      4
    AVNSP    3
    ADP      2
    ATC      2
    AMPH     2
    ADJ      2
    INT      4/
PARAMETER MAX(k) Number of classes scheduled for MOS k
  / ARTY     8
    TANK     5
    ADA     10
    AMO      7
    PAO      4
    MPO      8
    INFAN    8
    COMM     3
    SUP      5
    AIRSUP   5
    MT       4
    LOGS     3
    ENG      5
    FIN      2
    ADC      4
    AVNSP    4
    ADP      3
    ATC      3
    AMPH     2
    ADJ      3
    INT      4 /
PARAMETER MWAIT(k) Maximum wait between TBS and MOS classes
  / ARTY     9
    TANK    13
    ADA     11
    AMO     12
    PAO     11
    MPO      8
    INFAN    9
    COMM    17
    SUP     14
    AIRSUP  14
    MT      22
    LOGS    22
    ENG     10
    FIN     24
    ADC     12
    AVNSP   14
    ADP     22
    ATC     22
    AMPH    24
    ADJ     22
    INT     12 /
TABLE NEAREST(k,i) Wait from TBS to nearest follow on  MOS class
          TBS1  TBS2  TBS3  TBS4  TBS5  TBS6  TBS7  TBS8  TBS9
    ARTY    5     0     0     0     1     9     6     4     4
    TANK    7     2     8     3    13     1    10     2     4
    ADA     1     4     2     2     1     4     4     3     0
    AMO     4     2     2     5     0    12     1     7     0
    PAO     5     0     2     6     1     0     8     0     0
```

```
    MPO     8     3     5     0     2     0     5     0     0
    INFAN   0     0     0     0     0     0     9     0     0
    COMM   13     8     0    17    12     0     9     9     2
    SUP     0     8     0     7     2     5    14     0     5
    AIRSUP  0     8     0     5     0     5    14     0     5
    MT      0     8     0    22    17     5    14     0     5
    LOGS   13     8     0    22    17     5    14     0    18
    ENG     5     0     5     0     7     7     4     2    10
    FIN    18    13     5     0    24    12    21     7     0
    ADC     9     4     9     4    12     0     9     8     1
    AVNSP   5     0    10     5     0     5    14     0    10
    ADP     5     0    22    17    12     0     9    17    10
    ATC    13     8     0    22    17     5    14     0    18
    AMPH   18    13     5     0    24    12    21     7     0
    ADJ     5     0    22    17    12     0     9    17    10
    INT     5     0     5     0    12     0     9     7     0


PARAMETER WAITLEFT(j,k)     Waiting time for Leftovers              ;
     WAITLEFT(j,k)          = STMOS(k,j) - ENDTBS("TBS1");

SETS MPOS(j,k)   classes for MOS k
       XPOS(h,i)   Possible assignments from OCS to TBS
       YPOS(i,j,k) Possible assignments from TBS to MOS class
       SPILLPOS(j,k) Possible Leftover assignments to MOS class ;
       MPOS(j,k)                 = YES $ (ORD(j) LE MAX(k))              ;
       XPOS(h,i)$(TBSURO(i))  = YES $ (STTBS(i) LE STOCS(h)+14 AND
                                       STTBS(i) GE STOCS(h)+10)            ;
       YPOS(i,j,k)$(MPOS(j,k))= YES $ (STMOS(k,j) GE ENDTBS(i)+LEAVE(k)
        +OJT(k) AND STMOS(k,j) LE ENDTBS(i)+MWAIT(k)+OJT(k)+LEAVE(k)) ;
        SPILLPOS(j,k)$ (MPOS(j,k)) = YES$ (WAITLEFT(j,k) LT MAXLEFT);

SETS Y1POS(i,j,k)  Possible assignments from unrestricted TBS classes
       YWOPOS(i,j,k)   Possible assignments from warrant offr TBS class
       YWOPOS1(i,j,k) Possible mixed MOS assignments from WO TBS class;
       Y1POS(i,j,k)$(YPOS(i,j,k))= YES $ (ORD(i) NE W) ;
       YWOPOS(i,j,k)$(YPOS(i,j,k))= YES $ (ORD(i) EQ W) ;
       YWOPOS1(i,j,k)$(MOSWO(k))= YES $ (YWOPOS(i,j,k)) ;

PARAMETER WAITOCS(h,i)        Waiting time after OCS                ;
     WAITOCS(h,i) $ (XPOS(h,i))     = STTBS(i) - STOCS(h) - 10      ;
PARAMETER WAITTBS(i,j,k)      Waiting time after TBS                ;
     WAITTBS(i,j,k)$ (YPOS(i,j,k))  = STMOS(k,j) - ENDTBS(i)
                                      -  OJT(k) - LEAVE(k)          ;

PARAMETER MN(k) Preset minimum URO assignments from TBS class to MOS k
* Note :  Default MN(k) is approx 5% of annual quota
    /  ARTY          6
       TANK          1
       ADA           1
       AMO           1
       PAO           0
       MPO           0
       INFAN        12
       COMM          2
       SUP           2
       AIRSUP        2
       MT            2
       LOGS          2
       ENG           2
       FIN           2
       ADC           1
       AVNSP         1
       ADP           1
```

```
          ATC           1
          AMPH          1
          ADJ           0
          INT           0/
PARAMETER MIN(i,k)  Minimum URO assignments from TBS to MOS;
     MIN(i,k) $ (TBSURO(i))=  MN(k)     ;
*---------------------------------------------------------------------
*     OPTION SELECTION FOR MIN(k)
*---------------------------------------------------------------------
*  OPTIONS AVAILABLE :
*  OPTION 1 -- A minimum number to be assigned to every MOS from
*              each TBS class
*  OPTION 2 -- As in option 1, except the restriction is relaxed
*              for a TBS class if its assignment into a MOS exceeds
*              a preset limit determined by the user
*  OPTION 3 -- As in option 1, except the restriction is relaxed
*              for TBS classes selected by the user
*---------------------------------------------------------------------
*   Follow the instructions below to select and use a given
*   option:
*   (1) Enter the option number in the allocated space on the next line.
*       After this, if option 1 is selected, the option selection is
*       complete. Otherwise, go to step (2) if option 2 selected,
*       or step(3) if option 3 selected.
SCALAR OPT Index number of selected option/ 1/;

*   (2) Enter maximum allowable wait for an assignment
*       from TBS to any MOS class in the allocated space
*       below. After this, the option selection is complete.
SCALAR WMIN Maximum allowable wait for TBS to MOS assignment/ 4/;

*   (3) Enter 1 in the appropriate cell of Table MINOFF(k,i) below
*       if min(k) is to be relaxed for particular TBS class to MOS
*       combination. Otherwise, leave cell entry as zero.
TABLE MINOFF(k,i) Off switch for Minimum TBS to MOS assignment
           TBS1  TBS2  TBS3  TBS4  TBS5  TBS6  TBS7  TBS8  TBS9
     ARTY   0     0     0     0     0     0     0     0     0
     TANK   0     0     0     0     0     0     0     0     0
     ADA    0     0     0     0     0     0     0     0     0
     AMO    0     0     0     0     0     0     0     0     0
     PAO    0     0     0     0     0     0     0     0     0
     MPO    0     0     0     0     0     0     0     0     0
     INFAN  0     0     0     0     0     0     0     0     0
     COMM   0     1     1     0     0     1     0     0     0
     SUP    0     0     0     0     0     0     0     0     1
     AIRSUP 0     0     0     0     1     0     0     0     0
     MT     1     0     0     1     1     0     0     0     0
     LOGS   1     0     0     1     1     0     1     0     1
     ENG    0     0     0     0     0     0     0     0     1
     FIN    0     0     1     1     1     0     0     1     0
     ADC    0     0     0     0     0     0     0     0     0
     AVNSP  0     1     0     0     1     0     0     0     0
     ADP    1     0     0     1     1     0     0     0     0
     ATC    0     0     1     1     0     0     0     1     0
     AMPH   1     0     0     1     1     0     0     0     1
     ADJ    0     0     0     1     1     0     0     0     0
     INT    0     0     0     0     0     0     0     0     0

SET OP2(i,k) TBS to MOS combinations relaxed by option 2;
 OP2(i,k)$(NEAREST(k,i) GT WMIN) = YES$(OPT EQ 2);

SET OP3(i,k) TBS to MOS combinations relaxed by option 3;
 OP3(i,k)$(MINOFF(k,i) EQ 1) = YES$(OPT EQ 3);

 MIN(i,k) $ OP2(i,k) = 0;
 MIN(i,k) $ OP3(i,k) = 0;

VARIABLES
 GTBS(i)         Grd Z officers in TBS class i
 AX(h,i)         Air offrs assigned from OCS class h to TBS class i
 GX(h,i)         Grd offrs assigned from OCS class h to TBS class i
 Y(i,j,k)        Grd offrs from TBS i to class j(MOS k)
```

```
TBS(i)          Size of TBS class i
MOS(j,k)        Size of class j(MOS k)
SPILL(j,k)      Assignment of MOS k Leftovers to class j
INT(j,k)        Assignment of MOS k INT MOS to class j
VLAT(j,k)       Assignment of MOS k VOL LATMOVs to class j
DLAT(j,k)       Assignment of MOS k DIR LATMOVs to class j
FLTSEAT(j,k)    Assignment of MOS k FLT sources to class j
TDELAY          Total waiting time(in man weeks);

POSITIVE VARIABLES
 GTBS,AX,GX,Y,TBS,MOS,SPILL,INT,VLAT,DLAT,FLTSEAT;

EQUATIONS

AOCSBAL(h)           Balance eqn for AOCC h offrs
GOCSBAL(h)           Balance eqn for GOCC h offrs
TBSBAL(i)            Balance eqn0 for TBS i
TBS1BAL(i)           Balance eqn1 for TBS i
TBS2BAL              Balance eqn for TBS(warrant offr)
TBS3BAL(k)           Balance eqn for warrant officers in MOS k
MINASSIGN(i,k)       Minimum assignments from TBS i to MOS k
MOSDEF(j,k)          Definition of MOS
QUOTABAL(k)          Balance eqn for QUOTA k
BALLEFT(k)           Balance eqn for Leftover offrs
BALINT(k)            Balance eqn for Intended MOS offrs
BALVOL(k)            Balance eqn for VOLLAT offrs
BALDIR(k)            Balance eqn for DIRLAT offrs
BALFLT(k)            Balance eqn for FLTSEAT offrs
TDELAYDEF            Definition of total delay ;


AOCSBAL(h)..
     SUM(i$ XPOS(h,i),AX(h,i)) =E=  AOCCPASS(h) ;

GOCSBAL(h)..
     SUM(I$ XPOS(h,i),GX(h,i)) =E=  GOCCPASS(h) ;


TBSBAL(i) $ (TBSURO(i))..
     SUM(h $ XPOS(h,i),AX(h,i)) + SUM(H $ XPOS(h,i),GX(h,i))
                                   + GTBS(i) +ATBS(i)    =E=  TBS(i)    ;

TBS1BAL(i) $ (TBSURO(i))..
    TBS(i)  =E=  SUM((j,k) $ YPOS(i,j,k),Y(i,j,k))
                +    ATBS(i)     + SUM(H $ XPOS(h,i),AX(h,i));

TBS2BAL..
    TBS("TBS7")  =E=  SUM(k ,WOMOS(k)) ;
*  Change TBS7 if necessary to reflect correct warrant offr TBS class

TBS3BAL(k) $ (MOSWO(k))..
     SUM(j $ YPOS("TBS7",j,k),Y("TBS7",j,k))    =E= WOMOS(k)   ;
*  Change TBS7 if necessary to reflect correct warrant offr TBS class

MINASSIGN(i,k) $ (TBSURO(i))..
    SUM(j $ YPOS(i,j,k),Y(i,j,k))  =G= MIN(i,k)              ;


MOSDEF(j,k) $ (MPOS(j,k))..
    SUM(i $ Y1POS(i,j,k),Y(i,j,k)) +
      SUM(i $ YWOPOS1(i,j,k),Y(i,j,k)) + INT(j,k) + VLAT(j,k) +
          DLAT(j,k) +  FLTSEAT(j,k) + SPILL(j,k)  =E=    MOS(j,k) ;

QUOTABAL(k)..
    SUM(j $ MPOS(j,k),MOS(j,k))    =E=  QUOTA(k)   + LEFTOVER(k)  ;

BALLEFT(k)$(LEFTOVER(k) GT 0)..
    SUM(j $  MPOS(j,k),SPILL(j,k))    =E=  LEFTOVER(k)  ;


BALINT(k)$(INTTOT(k) GT 0)..
    SUM(j $  MPOS(j,k),INT(j,k))    =E=  INTTOT(k)   ;

BALVOL(k)$(VOLLAT(k) GT 0)..
    SUM(j $  MPOS(j,k),VLAT(j,k))    =E=  VOLLAT(k)   ;

BALDIR(k)$(INDLAT(k) GT 0)..
    SUM(j $  MPOS(j,k),DLAT(j,k))    =E=  INDLAT(k)   ;
```

```
BALFLT(k)$(FLT(k) GT 0)..
   SUM(j $  MPOS(j,k),FLTSEAT(j,k))  =E=  FLT(k)   ;
TDELAYDEF..
 TDELAY =E= SUM((h,i) $ XPOS(h,i),GX(h,i)*WAITOCS(h,i))+
              SUM((h,i) $ XPOS(h,i),AX(h,i)*WAITOCS(h,i))+
               SUM((i,j,k)$ Y1POS(i,j,k),Y(i,j,k)*WAITTBS(i,j,k))+
                SUM((i,j,k)$ YWOPOS1(i,j,k),Y(i,j,k)*WAITTBS(i,j,k))+
                 SUM((j,k)$SPILLPOS(j,k),SPILL(j,k)* WAITLEFT(j,k)) ;
* SET VARIABLE BOUNDS *
TBS.UP(i)$TBSURO(i)    =  TU      ;
TBS.LO(i)$TBSURO(i)    =  TL      ;
MOS.UP(j,k)$MOS5(k)    =  NMAX(k) ;
MOS.LO(j,k)$(ORD(j) LE NUM(k))
                           =  NMIN(k) ;
MOS.UP(j,k)$MOS3(k)    =  NSEAT(k,j) ;
FLTSEAT.UP(j,k)        =  CEIL(FLT(k)/NUM(k)) ;
* FIX VARIABLE VALUES
 SPILL.FX(j,k)$(WAITLEFT(j,k) GE MAXLEFT)  = 0;
 SPILL.FX(j,k)$(LEFTOVER(k) EQ 0)  = 0;
 FLTSEAT.FX(j,k)$(FLT(k) EQ 0)   = 0;
 INT.FX(j,k)$(INTTOT(k) EQ 0)    = 0;
 VLAT.FX(j,k)$(VOLLAT(k) EQ 0)   = 0;
 DLAT.FX(j,k)$(INDLAT(k) EQ 0)   = 0;

* OPTIONS :
OPTION LIMROW = 0;
OPTION LIMCOL = 0;
OPTION SOLPRINT = OFF;
OPTION DECIMALS = 0;
MODEL ASSIGNMENT/ALL/ ;
  SOLVE ASSIGNMENT USING LP MINIMIZING TDELAY;
  DISPLAY GTBS.L,AX.L,GX.L,Y.L,TBS.L,MOS.L,SPILL.L,INT.L,
   VLAT.L,DLAT.L,FLTSEAT.L,TDELAY.L
```

APPENDIX E

EXTRACT OF SOLUTION REPORT FROM ASSIGN GAMS PROGRAM

```
                  S O L V E      S U M M A R Y
     MODEL    ASSIGNMENT             OBJECTIVE  TDELAY
     TYPE     LP                     DIRECTION  MINIMIZE
     SOLVER   MINOS3                 FROM LINE  688

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS       1 OPTIMAL
**** OBJECTIVE VALUE               1033.0000

 RESOURCE USAGE, LIMIT           7.192       1000.000
 ITERATION COUNT, LIMIT        352           1000

 MINOS 3.4/ALTERED

 B. A. MURTAGH AND M. A. SAUNDERS,
 DEPARTMENT OF OPERATIONS RESEARCH,
 STANFORD UNIVERSITY,
 STANFORD   CALIFORNIA   94305   U.S.A.

 WORK SPACE NEEDED (ESTIMATE)     --    28896 WORDS.
 WORK SPACE AVAILABLE             --    28896 WORDS.
 (MAXIMUM OBTAINABLE              --   188444 WORDS.)

 EXIT -- OPTIMAL SOLUTION FOUND.

**** REPORT SUMMARY :          0      NONOPT
                                    0 INFEASIBLE
                                    0  UNBOUNDED

----     689 VARIABLE  GTBS.L          GRD Z OFFICERS IN TBS CLASS I

TBS1  40,     TBS2  91,     TBS3 157,     TBS4  56,     TBS5 117
TBS6 106,     TBS8  94,     TBS9 125

----     689 VARIABLE  AX.L            AIR OFFRS ASSIGNED FROM OCS CLASS
                                              H TO TBS CLASS I
```

	TBS1	TBS4	TBS8
OCS1	5		
CCS2		12	
OCS3			16

```
----     689 VARIABLE  GX.L            GRD OFFRS ASSIGNED FROM OCS CLASS
                                              H TO TBS CLASS I
```

	TBS1	TBS4	TBS8
OCS1	51		
OCS2		45	
OCS3			40

```
----     689 VARIABLE  Y.L             GRD OFFRS FROM TBS I TO
                                              CLASS J(MOS K)
```

	ARTY	TANK	ADA	AMO	PAO
TBS1.C1			4	2	
TBS2.C1	11	5			1
TBS2.C2	12		1	3	
TBS3.C3	12		4	3	
TBS4.C2		5			
TBS4.C4	16		4		
TBS5.C3					1
TBS5.C4				3	
TBS5.C5	25		3		

TBS5.C6	13			2
TBS6.C3		4		
TBS6.C8			1	
TBS7.C5				2
TBS8.C4		3		
TBS8.C7	12			
TBS8.C9			1	
TBS9.C5		6		
TBS9.C7				5
TBS9.C8	24			
TBS9.C10			1	

689 VARIABLE Y.L GRD OFFRS FROM TBS I TO CLASS J(MOS K)

+	MPO	INFAN	COMM	SUP	AIRSUP
TBS1.C1		15		31	17
TBS2.C2		29			
TBS3.C1			28		
TBS3.C2				27	7
TBS3.C3		29			
TBS4.C4		38			
TBS5.C3				13	13
TBS5.C5		29			
TBS6.C2			39		
TBS6.C6		30			
TBS6.C7	1				
TBS7.C4				5	
TBS8.C4				9	17
TBS8.C7		30			
TBS9.C3			18		
TBS9.C8		29			

+	MT	LOGS	ENG	FIN	ADC
TBS1.C1	22				
TBS2.C1			9		1
TBS3.C1		22			
TBS3.C2	20				
TBS4.C1				8	
TBS4.C2			20		4
TBS5.C3			7		
TBS6.C3					12
TBS7.C2				1	
TBS7.C3	7				
TBS8.C2		28			
TBS8.C3	11				
TBS8.C4			7		
TBS9.C2				25	
TBS9.C4					9

+	AVNSP	ADP	ATC	AMPH	ADJ
TBS2.C1	12	7			
TBS3.C1			5		
TBS4.C1				6	
TBS5.C2	8				
TBS6.C2		12			5
TBS7.C2		2			
TBS7.C3	8				
TBS8.C2			10		
TBS8.C3	6				
TBS9.C2				8	

689 VARIABLE Y.L GRD OFFRS FROM TBS I TO CLASS J(MOS K)

+	INT
TBS6.C3	2
TBS7.C3	12

---- 689 VARIABLE TBS.L SIZE OF TBS CLASS I

TBS1 150, TBS2 151, TBS3 186, TBS4 150, TBS5 206
TBS6 152, TBS7 37, TBS8 150, TBS9 150

---- 689 VARIABLE MOS.L SIZE OF CLASS J(MOS K)

	ARTY	TANK	ADA	AMO	PAO
C1	11	5	4	3	1
C2	12	5	2	3	2
C3	12	4	4	3	1
C4	16	3	4	3	1
C5	25	6	3	3	
C6	13		2	3	
C7	12		2	5	
C8	24		2		
C9			2		
C10			1		

+	MPO	INFAN	COMM	SUP	AIRSUP
C1	1	40	30	31	17
C2	2	30	40	29	13
C3	1	30	30	15	15
C4	1	39		15	17
C5	1	30			1
C6	2	30			
C7	1	30			
C8	1	30			

+	MT	LOGS	ENG	FIN	ADC
C1	22	25	9	16	6
C2	20	30	20	28	6
C3	20		7		12
C4			7		10

+	AVNSP	ADP	ATC	AMPH	ADJ
C1	14	10	7	6	22
C2	10	14	15	8	23
C3	15				

689 VARIABLE MOS.L SIZE OF CLASS J(MOS K)

+	INT
C1	22
C2	15
C3	17
C4	15

---- 689 VARIABLE SPILL.L ASSIGNMENT OF MOS K LEFTOVERS TO CLASS J

INFAN

ALL 0.

---- 689 VARIABLE INT.L ASSIGNMENT OF MOS K INT MOS TO CLASS J

ALL 0.

---- 689 VARIABLE VLAT.L ASSIGNMENT OF MOS K VOL LATMOVS TO CLASS J

	ADA	PAO	MPO	COMM	AIRSUP
C2		1	1		4
C3				10	
C4		1			
C6	1		1		

```
C7            1
C8                                    1

   +         MT         FIN         ADC         ADP         ATC

C1                        5           3           3
C2                                                            2
C3            2

   +        ADJ         INT

C2            5
C4                       10

----    689 VARIABLE  DLAT.L        ASSIGNMENT OF MOS K DIR LATMOVS TO
                                           CLASS J

              ADJ         INT

C1           10          18
C2                       11
C4                        1

----    689 VARIABLE  FLTSEAT.L     ASSIGNMENT OF MOS K FLT SOURCES TO
                                           CLASS J

             ADA         AMO         PAO         MPO       INFAN

C1                        1                       1
C2            1                       1           1           1
C3                                                1           1
C4                                                1           1
C5                        1                       1           1
C6            1           1                       1
C7            1
C8            1                                               1
C9            1

   +       COMM         SUP    AIRSUP          LOGS         FIN

C1            2                                   3           3
C2            1           2           2           2           2
C3            2           2           2
C4                        1
C5                                    1

   +        ADC       AVNSP         ATC         ADJ         INT

C1            2           2           2          12           4
C2            2           2           3          13           4
C3                        1                                   3
C4            1                                               4

----    689 VARIABLE  TDELAY.L      =          1033 TOTAL WAITING
                                                              TIME(IN MAN WEEKS)

**** FILE SUMMARY FOR USER 4562P

INPUT     ASSIGN   GAMS     A
OUTPUT    ASSIGN   LISTING  A

EXECUTION TIME       =        3.600 SECONDS
```

APPENDIX F

TIME CONVERSION CHART

Calendar Date	Model Date	Class with start date	Class with End date
23 Feb 87	Wk 1		
2 Mar 87	Wk 2		
9 Mar 87	Wk 3		
16 Mar 87	Wk 4		
23 Mar 87	Wk 5		
30 Mar 87	Wk 6		
6 Apr 87	Wk 7		
13 Apr 87	Wk 8		
20 Apr 87	Wk 9		
27 Apr 87	Wk 10		
4 May 87	Wk 11		
11 May 87	Wk 12		
18 May 87	Wk 13		
25 May 87	Wk 14		
1 Jun 87	Wk 15	TBS 1(5-87)	
8 Jun 87	Wk 16		
15 Jun 87	Wk 17		
22 Jun 87	Wk 18		
29 Jun 87	Wk 19		
6 Jul 87	Wk 20	TBS 2(6-87)	
13 Jul 87	Wk 21		
20 Jul 87	Wk 22		
27 Jul 87	Wk 23		
3 Aug 87	Wk 24		
10 Aug 87	Wk 25		
17 Aug 87	Wk 26	TBS 3(7-87)	
24 Aug 87	Wk 27		
31 Aug 87	Wk 28		
7 Sep 87	Wk 29		
14 Sep 87	Wk 30		
21 Sep 87	Wk 31	TBS 4(8-87)	
28 Sep 87	Wk 32		
5 Oct 87	Wk 33		
12 Oct 87	Wk 34		
19 Oct 87	Wk 35		
26 Oct 87	Wk 36	TBS 5(1-88)	
2 Nov 87	Wk 37		
9 Nov 87	Wk 38		
16 Nov 87	Wk 39		TBS 1(5-87)
23 Nov 87	Wk 40		
30 Nov 87	Wk 41		
7 Dec 87	Wk 42		
14 Dec 87	Wk 43		
21 Dec 87	Wk 44		TBS 2(6-87)
28 Dec 87	Wk 45		
4 Jan 88	Wk 46		
11 Jan 88	Wk 47		
18 Jan 88	Wk 48		
25 Jan 88	Wk 49		
1 Feb 88	Wk 50	TBS 6(2-88)	
8 Feb 88	Wk 51	TBS 7(WO-88)	TBS 3(7-87)
15 Feb 88	Wk 52		
22 Feb 88	Wk 53		
29 Feb 88	Wk 54		
7 Mar 88	Wk 55		
14 Mar 88	Wk 56	TBS 8(3-88)	
21 Mar 88	Wk 57		TBS 4(8-87)
28 Mar 88	Wk 58		
4 Apr 88	Wk 59		
11 Apr 88	Wk 60		

18 Apr 88	Wk 61		
25 Apr 88	Wk 62	TBS 9(4-88)	TBS 5(1-88)
2 May 88	Wk 63		
9 May 88	Wk 64		
16 May 88	Wk 65		TBS 7(WO-88)
23 May 88	Wk 66		
30 May 88	Wk 67		
6 Jun 88	Wk 68		
13 Jun 88	Wk 69		
20 Jun 88	Wk 70		
27 Jun 88	Wk 71		
7 Jul 88	Wk 72		
11 Jul 88	Wk 73		
18 Jul 88	Wk 74		TBS 6(2-88)
25 Jul 88	Wk 75		
1 Aug 88	Wk 76		
8 Aug 88	Wk 77		
15 Aug 88	Wk 78		
22 Aug 88	Wk 79		TBS 8(3-88)
29 Aug 87	Wk 80		
5 Sep 88	Wk 81		
12 Sep 88	Wk 82		
19 Sep 86	Wk 83		
26 Sep 83	Wk 84		
3 Oct 88	Wk 85		
10 Oct 88	Wk 86		TBS 9(4-88)
17 Oct 88	Wk 87		
24 Oct 88	Wk 88		
31 Oct 88	Wk 89		
7 Nov 88	Wk 90		
14 Nov 88	Wk 91		
21 Nov 88	Wk 92		
28 Nov 88	Wk 93		

APPENDIX G
USER INSTRUCTIONS

The user instructions are tailored for running both programs of the heuristic model on a personal computer since this is how it will be used at HQMC. The personal computer can be an IBM XT, AT or compatible and should have twin disk drives and a hard disk with at least 20 mb of memory installed. There are two parts in these instructions:

- Part I explains how to prepare the input file, define files and issue commands for execution of the program called SCHED.EXE. This program produces the MOS course schedules.
- Part II lists the inputs required for the second program called ASSIGN.GMS which solves the officer assignment problem. Then it gives the commands for program execution. Lastly, it explains how the output is to be intepreted.

The present version does not provide a software interface between the two programs, and requires output from the first program to be manually entered into the source code of the second program. This procedure is error-prone and should be considered as a temporary measure. Further software development to create the interface as well to enhance the overall user-friendliness of the package is strongly recommended.

1. PART I.

To run the FORTRAN program, the user must possess a diskette with the following files kept in a single directory: A.BAT, KEDIT.EXE, KEDIT352.DOC, INPUT.DAT and SCHED.EXE. The steps for using the program are described as follows:

- *Preparing the input file.* This file is called INPUT.DAT. It has four sections which are clearly annotated in the file. The input fields for file entries are given in the accompanying tables. As a convention, all data entries are right justified. For further clarification, refer to Appendix B which has a sample of the file.

TABLE 1
SECTION I -- OCS ENTRIES

Line	Column	Data input
3-5	3-5	OCS class start dates
3-5	13-15	Number of OCC Air officers in an OCS class

TABLE 2
SECTION II -- TBS ENTRIES

Line	Column	Data input
9-16	3-5	TBS class end dates (unrestricted officer class only)
9-16	13-15	TBS class start dates (unrestricted officer class only)
9-16	23-25	Number of non-OCC Air officers in a TBS class[8]
20	5	Warrant officer TBS class number
20	13-15	Warrant officer TBS class end date

[8]HQMC uses the synonym Z officers to describe non-OCC officers

TABLE 3
SECTION III--MOS ENTRIES

Line	Column	Data input
23-43	3-5	MOS course duration
23-43	8-10	Minimum MOS assignment from each TBS class (using five percent of MOS assignments from TBS)
23-43	13-15	MOS output from TBS classes
23-43	18-20	MOS output from non-TBS sources
23-43	23-25	Number of Warrant officers in MOS with mixed MOS classes
23-43	28-30	Length of OJT after TBS
23-43	33-35	Length of enforced leave after TBS
23-43	38-40	Ending date of last MOS class scheduled in the previous year
23-43	43-45	Number of MOS classes per year (enter 99 for USMC MOS's)
23-43	50	Selection indicator for preselected class dates
23-43	53-55	Minimum MOS class size (enter 99 for non USMC MOS's)
57-62	3-55	MOS Class start dates (non USMC MOS's only)
64-69	3-55	MOS Class seats (non USMC MOS's only)

TABLE 4
SECTION IV--PRESELECTED COURSE ENTRIES

Line	Column	Data input
73-77	3-5	Number of MOS classes held per year
73-77	8-55	Preselected MOS course schedule

To edit file INPUT.DAT, type:

KEDIT INPUT.DAT

After the entries are made, type the following command to save:

FILE

- *Defining files* . File definition is necessary because the operating system must know where to look for input files that feed the FORTRAN program, as well as where to send output. The file definition for the input file and two output files generated by the program is made by executing the batch file A.BAT. The next command to be typed is simply:

  A

- *Executing SCHED.EXE.* Now, the program is ready for execution. To do this, type:

  SCHED

  Upon successful execution, two output files are produced--SCHED.GMS and USER.OUT. These will be copied on to the diskette in the current drive automatically. The format for these files were explained in chapter three. At this point, get a hard copy of the output file SCHED.GMS since it contains information required to run the second program.

2. PART II

To run the second program, the user must possess a data diskette with the following files: SETPATH.BAT, KEDIT.EXE, KEDIT352.DOC, and ASSIGN.GMS. He must also have the GAMS modeling software. The PC configuration should be set up so that DOS is installed on the A drive, the data diskette on the B drive and the GAMS files placed in the same directory on the C drive(hard disk). The steps for using this program are described as follows:

- *Preparing the GAMS program for execution.*

  (a) Before the GAMS program can be executed, it is necessary to update its data contents to correctly reflect those of the FY being planned. To modify the program, first type:

  KEDIT ASSIGN.GMS

  The computer then responds by displaying the first page of the GAMS program on the terminal.

  (b) The next step is to enter the new data. The parameters/tables requiring update are:

  - `PARAMETER QUOTA Yearly output quota for MOS`
  - `PARAMETER LEFTOVER TBS output brought fwd from previous yr`
  - `PARAMETER WOMOS WOs to be assigned to mixed MOS classes`
  - `SCALAR W Warrant officer TBS class number`
  - `SCALAR OCSATT OCS attrition rate`
  - `SCALAR MAXLEFT Maximum waiting time allowed for Leftovers`
  - `SCALAR TL Lower capacity of unrestricted offr TBS class`

- SCALAR TU Upper capacity of unrestricted offr TBS class
- PARAMETER LEAVE Enforced leave after TBS
- PARAMETER OJT On-the-Job training period after TBS
- PARAMETER FLT Number of Grounded Air officers
- PARAMETER INTTOT Number of Intended MOS officers
- PARAMETER VOLLAT Number of Vol LATMOV officers
- PARAMETER INDLAT Number of Directed LATMOV officers
- PARAMETER GOCC Number of Grd OCC officers in each OCS class
- PARAMETER AOCC Number of Air OCC Officers in each OCS class
- PARAMETER ATBS Number of non OCC Air officers in each TBS class
- TABLE NSEAT Reserved seats at non-USMC MOS schools
- PARAMETER NMIN Lower capacity for MOS class
- PARAMETER STOCS Starting week of OCS class
- PARAMETER STTBS Starting week of TBS class
- PARAMETER ENDTBS Ending week of TBS class
- TABLE STMOS Starting week of MOS class
- PARAMETER NUM Number of MOS classes per year
- PARAMETER MAX where MAX equals NUM if the last MOS class start before the last TBS class ends. Otherwise, MAX = NUM + 1
- PARAMETER MWAIT Maximum waiting time from a TBS class to a follow on MOS class
- TABLE NEAREST Waiting time from a TBS class to follow on MOS class

The values for the last five inputs, *STMOS*, *NUM*, *MAX*, *MWAIT* and *NEAREST* are extracted from the SCHED.GMS output file produced in Part I. The remaining input values have to be obtained from planning documents available at HQMC.

Now, go ahead and enter the values for the above data in the designated input fields within the GAMS program.

(c) Perform the next step by scrolling the screen to the program area where the equations are shown. Examine the two equations:

```
TBS2BAL..
   TBS("TBS7")  =E=  SUM(k ,WOMOS(k))
```

and

```
TBS3BAL(k) $ (MOSWO(k))..
```

```
SUM(j $ YPOS("TBS7",j,k),Y("TBS7",j,k)) =E= WOMOS(k)
```

Check if the number entered in parentheses correspond to the class number for the Warrant Officer TBS class. If not, change the number to correctly reflect the Warrant Officer TBS class number.

(d) Finally, a suitable option for *min*(k) has to be selected. *Min*(k) is the minimum assignment to MOS k from each TBS class. Three options are available:

- *Option one* : a minimum number to be assigned to every MOS from each TBS class.
- *Option two* : as in option one, except the restriction is relaxed for TBS class to MOS class assignments whose waiting time exceeds a preset limit.
- *Option three* : as in option one, except the restriction is relaxed for selected TBS classes.

Detailed instructions for selecting an option are written in the program. The following is a printout from the relevant program area:

```
*-----------------------------------------------------------------
*     OPTION SELECTION FOR MIN(k)
*-----------------------------------------------------------------
* OPTIONS AVAILABLE :
* OPTION 1 -- A minimum number to be assigned to every MOS from
*             each TBS class
* OPTION 2 -- As in option 1, except the restriction is relaxed
*             for a TBS class if its assignment into a MOS exceeds
*             a preset limit determined by the user
* OPTION 3 -- As in option 1, except the restriction is relaxed
*             for TBS classes selected by the user
*-----------------------------------------------------------------
* Follow the instructions below to select and use a given
* option:
* (1) Enter the option number in the allocated space on the next line.
*     After this, if option 1 is selected, the option selection is
*     complete. Otherwise, go to step (2) if option 2 selected,
*     or step(3) if option 3 selected.
SCALAR OPT Index number of selected option/ 1/;

* (2) Enter maximum allowable wait for an assignment
*     from TBS to any MOS class in the allocated space
*     below. After this, the option selection is complete.
SCALAR WMIN Maximum allowable wait for TBS to MOS assignment/ 4/;

* (3) Enter 1 in the appropriate cell of Table MINOFF(k,i) below
*     if min(k) is to be relaxed for particular TBS class to MOS
*     combination. Otherwise, leave cell entry as zero.
TABLE MINOFF(k,i) Off switch for Minimum TBS to MOS assignment
            TBS1  TBS2  TBS3  TBS4  TBS5  TBS6  TBS7  TBS8  TBS9
     ARTY   0     0     0     0     0     0     0     0     0
     TANK   0     0     0     0     0     0     0     0     0
     ADA    0     0     0     0     0     0     0     0     0
     AMO    0     0     0     0     0     0     0     0     0
     PAO    0     0     0     0     0     0     0     0     0
     MPO    0     0     0     0     0     0     0     0     0
     INFAN  0     0     0     0     0     0     0     0     0
     COMM   0     1     1     0     0     1     0     0     0
     SUP    0     0     0     0     0     0     0     0     1
     AIRSUP 0     0     0     0     1     0     0     0     0
     MT     1     0     0     1     1     0     0     0     0
     LOGS   1     0     0     1     1     0     1     0     1
```

ENG	0	0	0	0	0	0	0	0	1
FIN	0	0	1	1	1	0	0	1	0
ADC	0	0	0	0	0	0	0	0	0
AVNSP	0	1	0	0	1	0	0	0	0
ADP	1	0	0	1	1	0	0	0	0
ATC	0	0	1	1	0	0	0	1	0
AMPH	1	0	0	1	1	0	0	0	1
ADJ	0	0	0	1	1	0	0	0	0
INT	0	0	0	0	0	0	0	0	0

Select an appropriate option by following the above instructions.

- *Execution of the GAMS program*

  (a) Type the next command to direct DOS to check the correct drives for program executable files:

  SETPATH

  (b) The system is now ready for execution of the GAMS program. To execute, type the command:

  GAMS ASSIGN

  Assuming an optimal solution is obtained, after several minutes, the screen will display that an optimal solution is found and the results are being written to scratch files. This means that the output is being collected in a file called ASSIGN.LST which will be automatically copied on to the data diskette in drive B. To look at these results, type:

  KEDIT ASSIGN.LST

- *Intepretation of the output* . GAMS provides a very detailed output. The best way to study this output is through a hard copy printout. Look at the printout from the following section onwards:

```
                S O L V E       S U M M A R Y
      MODEL   ASSIGNMENT          OBJECTIVE  TDELAY
      TYPE    LP                  DIRECTION  MINIMIZE
      SOLVER  MINOS3              FROM LINE  748

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS       1 OPTIMAL
**** OBJECTIVE VALUE              1033.0000
```

The last three lines in the above section shows the outcome of the solution effort. The solver status of NORMAL COMPLETION says there was no abnormal interruption during the solution process. The model status of OPTIMAL shows the solution is globally optimal with an objective value of 1033 as reported in the third line.

The next section is the solution display which starts at the line:

```
---- 689 VAR GTBS.L       GRD Z OFFICERS IN TBS CLASS I
```

All necessary information for making the following year's officer assignments is contained in the solution display. This can be verified by checking through its entire contents. The solution display, like the rest of GAMS output, is self-documenting. Every variable is displayed side by side with their definition. For first time users, however, the following example may help to clarify interpretation of the results.

This example explains the variable display format. Consider the variables GX which are the assignments of Ground officers from OCS to TBS. In the sample display below for variables GX, the number of assignments from OCS class 1 to TBS class 1 is 51, that from OCS class 2 to TBS class 4 is 45, and from OCS class 3 to TBS class 8 it is 40 :

```
---- 689 VARIABLE  GX.L           GRD OFFRS ASSIGNED FROM OCS CLASS
                                       H TO TBS CLASS I

              TBS1         TBS4         TBS8

OCS1           51
OCS2                        45
OCS3                                     40
```

LIST OF REFERENCES

1. Plotnicki, W.J., Garfinkel, R.S., "Selecting Academic Courses to Maximise Student Flow: A Simulation Approach," *Socio Econ Planning Science*, v.20, No.4, pp. 193-199, 1986.

2. Bisschop, J., Meeraus, A., "Selected Aspects of a General Algebraic Modeling Language," *Optimization Techniques : Proceedings of the 9th IFIP Conference on Optimization Techniques*, Part 2, pp. 223 - 233, 1980.

3. Bisschop, J., Meeraus, A., "Toward Successful Modeling Applications in a Strategic Planning Environment," *Large Scale Linear Programming*, pp. 711 - 745, 1981.

4. Kendrick, D., Meeraus, A., "GAMS : An Introduction," Development Research Department, The World Bank, Washington, D.C. 20433, 1987 (revised).

5. Rosenthal, R.E., "Review of GAMS/MINOS Modeling Language and Optimization Program," *OR/MS Today*, V. 13, No.3, pp. 24 - 32, 1986.

6. Rosenthal, R.E., "Tutorial on the GAMS Modeling Language," Operations Research Department, Naval Postgraduate School, Monterey, CA 93943, 1987.

7. Fisher, M.L., "The Lagrangean Relaxation Method for Solving Integer Programming Problems," *Management Science*, v. 27, pp. 1 - 17, January 1981.

8. Fisher, M.L., "An applications Oriented Guide to Lagrangean Relaxation," *Interfaces*, v. 15, pp. 10 - 21, 2 March-April 1985.

9. Veinott, A.F., Wagner, H.M., "Optimal Capacity Scheduling," *Operations Research*, v. 10, pp. 518 - 532, 1962.

10. Rosenthal, R.E., "A Non-linear Flow Algorithm for Maximization of Benefits in a Hydoelectric Power System," *Operations Research*, v. 29, pp. 763 - 786, July-August 1981.

11. Benders, J.F., "Partitioning for Solving Mixed Variable Programming Problems," *Numerische Mathematik 4*, pp. 238 - 252, 1962.

12. Van Roy, T.J., "Cross Decomposition for Mixed Integer Programming," *Mathematical Programming*, v.25, pp. 46 - 63, 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey,CA 93943	1
4.	Professor Richard E. Rosenthal, Code 55Rl Department of Operations Research Naval Postgraduate School Monterey,CA 93943	5
5.	Professor Paul R. Milch, Code 55Mh Department of Operations Research Naval Postgraduate School Monterey,CA 93943	5
6.	LTC G. C. Axtell, Code MPI-40 Headquarters United States Marine Corps Washington,D.C. 20380-0001	1
7.	LTC P. R. Stenner, Code MMOA-3 Headquarters United States Marine Corps Washington,D.C. 20380-0001	1
8.	Head of Training Information Section, Code TPI-60 Headquarters United States Marine Corps Washington,D.C. 20380-0001	1
9.	MAJ D. Hundley, Code MMOA-3 Headquarters United States Marine Corps Washington,D.C. 20380-0001	1
10.	MAJ R. Huck, Code MMOA-3 Headquarters United States Marine Corps Washington, D.C. 20380-0001	1
11.	MAJ K. Riecks, Code MPI-45 Headquarters United States Marine Corps Washington,D.C. 20380-0001	1

12. Dr Alexander Meeraus | 1
The World Bank
1818 H Street N.W.
Washington,D.C. 20433

13. MAJ Chng Keng Seng | 3
530 East Coast Road #04-04
Singapore, 1545
Republic of Singapore

END

12-87

DTIC